Formulation for Planar Multibody Dynamics

Prepared by Paulo Flores and Hamid M. Lankarani, 2012

University of Minho
Mechanical Engineering Department
Campus de Azurém
4800-058 Guimarães, Portugal
Phone: 253 510220
E-mail: pflores@dem.uminho.pt
URL: www.dem.uminho.pt/People/pflores

Wichita State University
Mechanical Engineering Department
1845 Fairmount
Wichita KS 67260-133, USA
Phone: 316 9783236
E-mail: hamid.lakarani@wichita.edu
URL: www.engr.wichita.edu/hlankarani

Objective and Embryogenesis of Multibody Systems

The main objective of this book is to present a complete and suitable methodology for dynamic analysis of multibody systems.

As the name suggests, a multibody system is an assembly of several bodies connected to each other by joints and acted upon by forces.

A body, that can be rigid or flexible, is composed by a collection of material points.

A joint allows for certain degrees of freedom and constrains others. In practice, joints are connection devices such as bearings, rod guides, etc., which from mathematical point of view are denominated as revolute joints, translational joints, etc., according to the relative degrees of motion permitted.

The forces can have different sources and different levels of complexity.

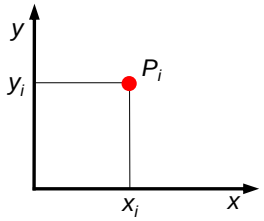


Fig. 1 Free particle

Fig. 2 Newton

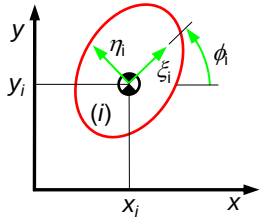


Fig. 3 Free rigid body

Fig. 4 Euler

Multibody systems dynamics is related to classical and analytical mechanics. The most simple element of a multibody system is a free particle introduced by Newton (1686) in his "*Philosophiae Naturalis Principia Mathematica*" or simply "*Principia*". The essential element, the rigid body, was defined by Euler (1776) in his contribution entitled "*Nova methodus motum corporum rigidarum determinandi*". For the modeling of constrains and joint, Euler already used the free body principle resulting in reaction forces.

The equations obtained are known in multibody dynamics as Newton-Euler equations of motion, also called translational and rotational equations of motion.

Embryogenesis of Multibody Systems and Formulation

In fact, dynamics of multibody systems is based on classical mechanics and has a long and prolific history. A free or unconstrained material point is the simplest multibody system that can be studied by applying the equations of motion established with genial acumen by Newton (1686).

A system of constrained rigid bodies was considered by d'Alembert (1743) in his "*Traité de Dynamique*" where he distinguished between free and hinder components of the motion. A systematic analysis of constrained multibody systems was developed by Lagrange (1788), combining the d'Alembert fundamental idea with the principle of virtual work (Greenwood 1965).

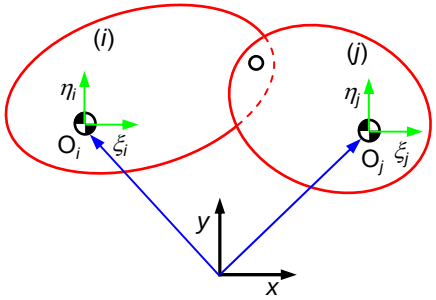


Fig. 5 Constrained MBS

Fig. 6 d'Alembert

Fig. 7 Lagrange

Fig. 8 Nikravesh

However, it was only during the last half of century that multibody dynamics received a conspicuous attention thanks to the impressive computer progresses verified at both software and hardware levels. As a consequence, a great variety of methodologies have been proposed, despite the fact that all of them can be derived from a few fundamental principles of mechanics (Nikravesh 1988).

Multibody systems serve as a basis for many models of mechanical systems and have been applied in many areas of science and engineering. Multibody systems are often used to analyze biological and human locomotion. Controlled systems are frequently prototyped through computer simulation of multibody models. There are also applications in medical, robotics, space subsystems and computer games.

Formulation and Structure of the Work

The formulation of multibody system dynamics adopted in this work uses the generalized absolute coordinates and the Newton-Euler approach to derive the equations of motion. This formulation results in the establishment of a mixed set of partial differential and algebraic equations, which are solved in order to predict the dynamic behavior of multibody systems. The Newton-Euler approach is very straightforward in terms of assembling the equations of motion and providing all joint reaction forces. Additionally, the equations of motion are solved by using the Baumgarte stabilization technique with the intent of keeping the constraint violations under control.

$$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_{\mathbf{q}}^{T} \\ \mathbf{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{g} \\ \boldsymbol{\gamma} - 2\alpha\dot{\mathbf{\Phi}} - \beta^{2}\mathbf{\Phi} \end{Bmatrix}$$

This work contains eight main sections that are organized as follows:

1 Introduction

2 Multibody System Concept and Applications

3 Types of Coordinates, Global and Local Coordinates

4 Kinematic Analysis and Constraint Equations

5 Numerical Methods in Kinematic Analysis

6 Equations of Motion for Multibody Systems

7 Numerical Methods in Dynamic Analysis

8 Baumgarte Stabilization Method

Multibody System Concept

In a simple way, it can be said that a general multibody system embraces two main characteristics, namely:

(*i*) mechanical components that describe large translational and rotational displacements and

(*ii*) kinematic joints that impose some constraints or restrictions on the relative motion of the bodies.

In other words, a multibody system (MBS) encompasses a collection of rigid and/or flexible bodies interconnected by kinematic joints and possibly some force elements (Nikravesh 1988, 2008).

Driving elements and prescribed trajectories for given points of the system components, can also be represented under this general concept of multibody system.

Figure 9 depicts an abstract representation of a multibody system (Flores et al. 2008).

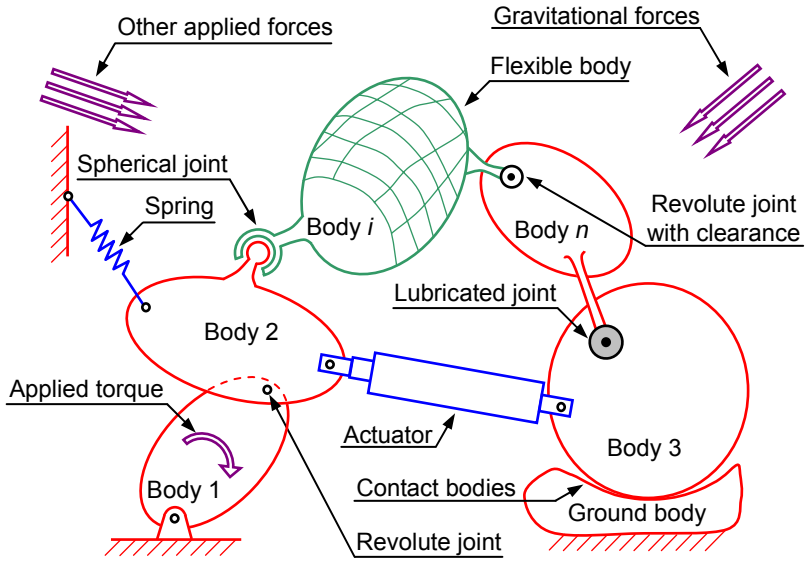


Fig. 9 Abstract representation of a multibody system with its most significant components: bodies, joints and forces elements

Multibody System Concept

The bodies that belong to a multibody system can be considered as rigid or flexible.

A body is said to be rigid when its deformations are assumed to be small such that they do not affect the global motion produced by the body. In the two-dimensional space, the motion of a free rigid body can be fully described by using three generalized coordinates associated with the three degrees of freedom.

In turn, when a body includes some amount of flexibility, it has three rigid degrees of freedom plus the number of generalized coordinates necessary to describe the deformations (Shabana 1989).

The expression flexible multibody system refers to a system holding deformable bodies with internal dynamics.

In fact, rigid bodies are a representation of reality because bodies are not absolutely rigid in nature. However, in a good number of common applications, the bodies are significantly stiff and, therefore, their flexibility can be disregarded and the bodies can be considered to be perfectly rigid. Within the scope of the present work, only rigid bodies are considered.

Fig. 10 Rigid multibody system

Fig. 11 Flexible multibody system

Multibody System Concept

By and large, the kinematic joints that can exist in multibody systems constrain the relative motion between the bodies connected by them. Two of the most typical kinematic joints employed in multibody systems are the revolute and the translational joints (Haug 1989).

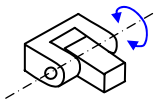


Fig. 12 Revoute joint

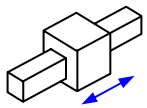


Fig. 13 Translational joint

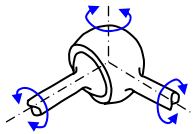


Fig. 14 Spherical joint

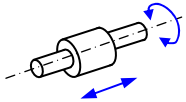


Fig. 15 Cilindrical joint

While the force elements represent the internal forces that are produced in the system and they are associated with the relative motion of the bodies. The forces applied over the multibody system components can be the result of springs, dampers, actuators or external forces. External applied forces of different nature and different level of complexity can act on a multibody system with the purpose to simulate the interactions among the system components and between these and the surrounding environment (Schiehlen 1990).

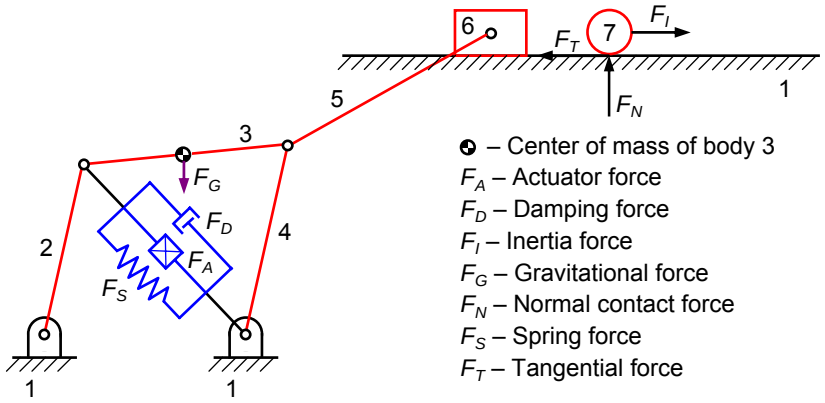
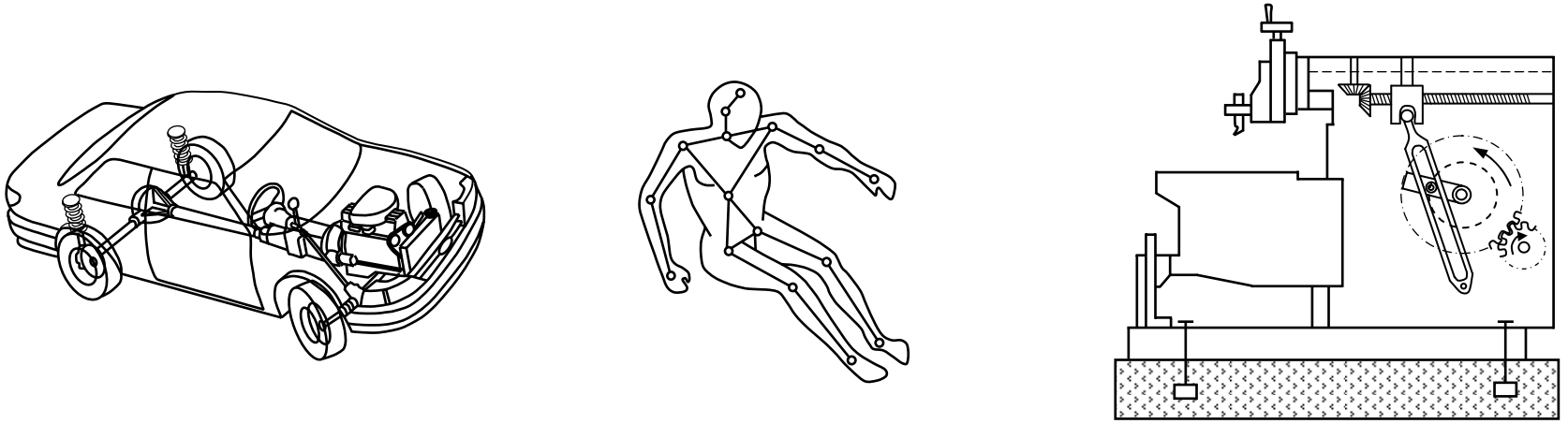


Fig. 16 Different types of forces that can be present in a mechanical system

Application Examples

A multibody system can be used to study the kinematic and dynamic motion characteristics of a wide variety of systems in a large number of engineering fields of application. Multibody systems can range from very simple to highly complex. There is no doubt that multibody systems are ubiquitous in engineering and research activities, such as robotics (Zhu et al. 2006), heavy machinery (Seabra et al. 2007), automobile vehicles (Ambrósio and Veríssimo 2009), biomechanics (Silva and Ambrósio 2002), mechanisms (and Claro 2007a), railway vehicles (Pombo and Ambrósio 2008), space systems (Ambrósio et al. 2007).

Figure 17 shows some multibody system examples of application, which result from the association of structural and mechanical subsystems with the purpose to transmit or transform a given motion.



(a) (b) (c)

Fig. 17 Examples of application of multibody systems: (a) automobile vehicle model; (b) human biomechanical model; (c) shaping machine

In a simple manner, multibody systems methodologies include the following two phases: (*i*) development of mathematical models of systems and (*ii*) implementation of computational procedures to perform the simulation, analysis and optimization of the global motion produced (Haug 1989).

Degrees of Freedom

Prior to establish the equations of motion that govern the dynamic behavior of multibody systems, it is first necessary to select the way how to describe them.

The description variables must be able to characterize, at any instant of time, the configuration of the system, that is, the position of all the material points that compose the bodies. The description variables, also called generalized coordinates, must uniquely define the position of the system components at any instant of time during the multibody system analysis. The expression generalized coordinates is employed to include both linear and angular coordinates (Huston 1990).

The minimum number of variables necessary to fully describe the configuration of a system is denominated as degrees of freedom (DOF) of the system, or simply mobility (Müller 2009). When the configuration of a multibody system is completely defined by the orientation of one of its bodies, the system is said to have one degree of freedom.

The number of degrees of freedom can also be defined as the number of independent generalized coordinates required to uniquely describe the configuration of a system.

The number of degrees of freedom of a multibody system can be evaluated as the difference between the system coordinates and the number of independent constraints. For planar multibody systems, the mathematical expression that summarizes this concept is known as the Grüebler-Kutzback criterion and is written as (Shigley and Uicker 1995)

$$DOF = 3n_b - m \tag{1}$$

where n_b represents the number of rigid bodies that compose the multibody system and m is the number of independent constraints.

Degrees of Freedom and Types of Coordinates

For example, the planar four-bar mechanism shown in Fig. 18 has four bodies, eight revolute joint constraints and three ground body constraints, yielding one degree of freedom. Figure 19 shows a triple pendulum that comprises four bodies, six revolute joint constraints and three ground body constraints, which results in three degrees of freedom.

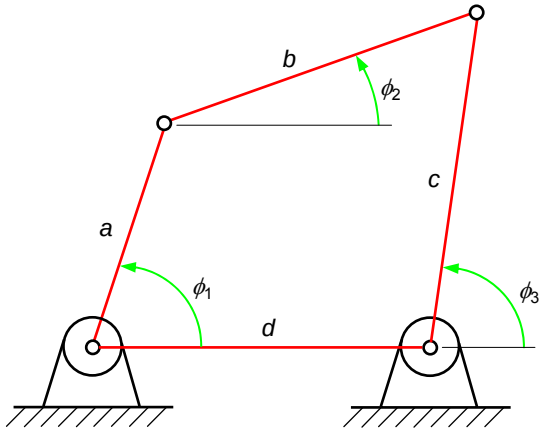


Fig. 18 Four-bar mechanism

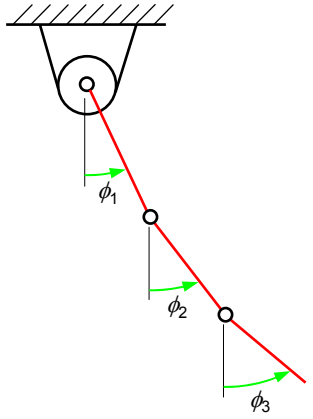


Fig. 19 Triple pendulum

It is not unanimous and it is not a simple task either to define a criterion to classify the different types of coordinates that can be used to describe the configuration of multibody systems.

A general and broad embracing rule to group the generalized coordinates is to divide them into independent and dependent coordinates (Wehage and Haug 1982).

The independent coordinates are free to vary arbitrarily, while the dependent coordinates are required to satisfy the equations of constraints.

Additionally, the dependent coordinates are classified as absolute coordinates (Orlandea et al. 1977), relative coordinates (Chace 1967) and natural coordinates (Jálon and Bayo 1994).

Types of Coordinates

Figure 20 summarizes the different types of coordinates most frequently used to describe the configuration of multibody systems. Other possible way to classify the coordinates is to split them into Lagrangian and Eulerian coordinates.

According to Nikravesh (1998), the general distinction between Lagrangian and Eulerian coordinates is that the former allows the definition of the position of a body relative to a moving coordinate system, whereas the later normally requires that the position of each body in space be defined relative a fixed global coordinate system.

Therefore, Eulerian coordinates require that a large number of coordinates be defined to specify the position of each body of a multibody system.

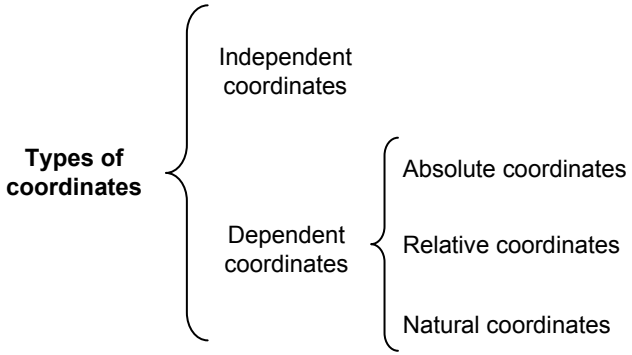


Fig. 20 Types of coordinates frequently used in multibody systems formulation

In the present work, the vector of coordinates **q** is defined as a column vector that contains all the variables used in the description of the configuration of multibody systems.

In the following paragraphs, the different types of coordinates are briefly presented by using simple demonstrative examples.

Types of Coordinates

Let consider the triple pendulum illustrated in Fig. 19, from which it can be observed that the configuration of the system can uniquely and completely be defined if the angular variables ϕ_1, ϕ_2 and ϕ_3 are known for each instant of time during the analysis. This fact is not surprising because the triple pendulum has three degrees of freedom in the measure that it has three independent motions, each one associated with each arm rotation. In this case, the set of generalized coordinates can be expressed as

$$\mathbf{q} = \{\phi_1 \quad \phi_2 \quad \phi_3\}^T \tag{2}$$

where the superscript *T* represents the transpose mathematical operation.

Similarly, in the four-bar mechanism presented in Fig. 18 there is a set of three angular variables ϕ_1, ϕ_2 and ϕ_3 that define the configuration of the system. However, these three variables are not independent because the system has only one degree of freedom. The angular variables can be related to each other by writing two algebraic equations of the closed kinematic chain associated with the four-bar mechanism, yielding

$$a\cos\phi_1 + b\cos\phi_2 - c\cos\phi_3 - d = 0 \tag{3}$$

$$a\sin\phi_1 + b\sin\phi_2 - c\sin\phi_3 = 0 \tag{4}$$

in which *a, b*, *c* and *d* are the lengths of the bodies. Equations (3) and (4) can be numerically solved for ϕ_2 and ϕ_3 as function of ϕ_1. Thus, ϕ_1 is the only variable required to be known to completely define the configuration of the four-bar mechanism. The vector of coordinates for this four-bar mechanism is written as

$$\mathbf{q} = \{\phi_1 \quad \phi_2 \quad \phi_3\}^T \tag{5}$$

Types of Coordinates

In short, it can be said that for the case of independent coordinates, the variables used are those associated with the degrees of freedom of the system.

While for the dependent coordinates, besides the variables associated with the degrees of freedom, it is also required to consider other set of coordinates and the necessary constraint equations that relate the independent and dependent coordinates.

The formulation of multibody systems with independent coordinates produces equations with a high degree of nonlinearity, being hard their computational implementation.

It must be highlighted that in some cases, the independent coordinates do not define the system configuration in a univocal manner.

Figure 21 illustrates the case where for the same value of the angle ϕ_1, the four-bar mechanism can assume two different configurations (Flores and Claro 2007b).

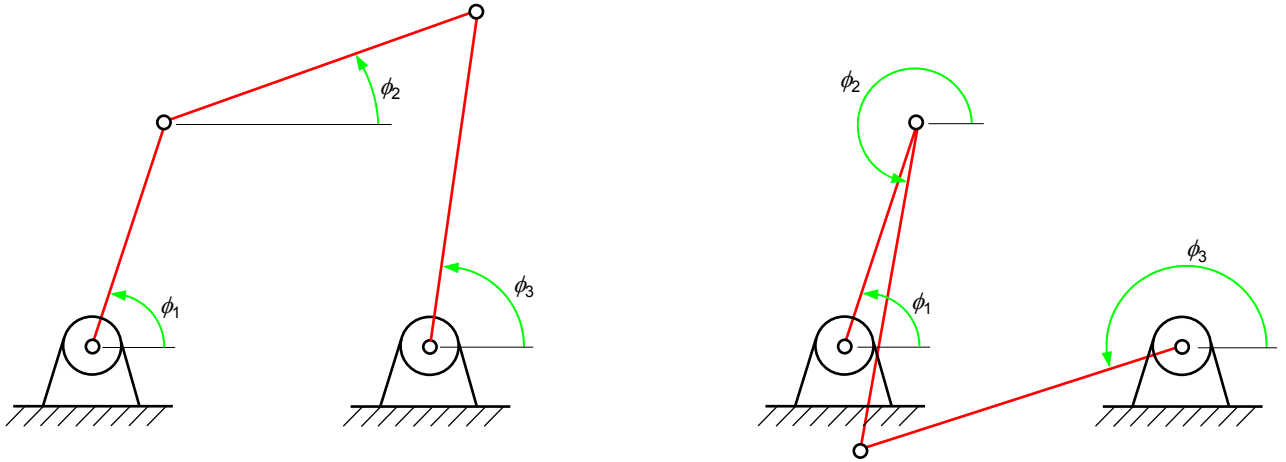


Fig. 21 Two different four-bar mechanism configurations for the same angle ϕ_1

Absolute Coordinates

Let consider the four-bar mechanism of Fig. 22 to discuss the main features of the absolute coordinates, in which the center of mass of each body is represented.

In the absolute coordinates, also called reference point coordinates, the generalized coordinates define the position of each body (typically the location of the center of mass) and the orientation or angle of the body in the system.

In the planar case, this situation corresponds to three variables, two Cartesian coordinates (x and y) and one angle (ϕ) with respect to a global coordinate system.

The angle must be measured in the counter clockwise direction and with respect to x-axis.

Thus, for the case of the four-bar mechanism represented in Fig. 22, the vector of generalized coordinates is expressed by

$$\mathbf{q} = \{x_1 \quad y_1 \quad \phi_1 \quad x_2 \quad y_2 \quad \phi_2 \quad x_3 \quad y_3 \quad \phi_3\}^T \tag{6}$$

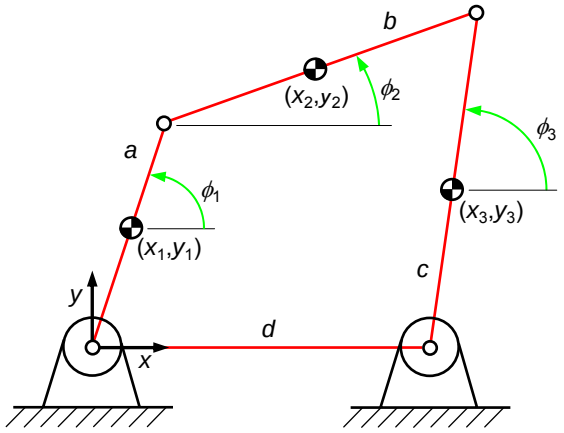


Fig. 22 Four-bar mechanism described by absolute coordinates

Absolute Coordinates

It is obvious that these nine coordinates are not independent because the system has only one degree of freedom. Therefore, it is necessary to write eight algebraic constraint equations, which can easily be obtained from the information contained in Fig. 22, yielding

$$x_1 - \frac{a}{2}\cos\phi_1 = 0 \tag{7}$$

$$y_1 - \frac{a}{2}\sin\phi_1 = 0 \tag{8}$$

$$x_1 + \frac{a}{2}\cos\phi_1 - x_2 + \frac{b}{2}\cos\phi_2 = 0 \tag{9}$$

$$y_1 + \frac{a}{2}\sin\phi_1 - y_2 + \frac{b}{2}\sin\phi_2 = 0 \tag{10}$$

$$x_2 + \frac{b}{2}\cos\phi_2 - x_3 - \frac{c}{2}\cos\phi_3 = 0 \tag{11}$$

$$y_2 + \frac{b}{2}\sin\phi_2 - y_3 - \frac{c}{2}\sin\phi_3 = 0 \tag{12}$$

$$x_3 - \frac{c}{2}\cos\phi_3 - d = 0 \tag{13}$$

$$y_3 - \frac{c}{2}\sin\phi_3 = 0 \tag{14}$$

Absolute Coordinates and Relative Coordinates

Thus, for any specified configuration, that is, by knowing any of the nine coordinates, the remaining eight variables can be obtained by solving the set of eight nonlinear algebraic equations in eight unknowns. This procedure is usually performed by employing a numerical algorithm, such as the Newton-Raphson method (Eich-Soellner and Führer 1998).

In summation, the formulation of multibody systems with absolute coordinates is quite simple and straightforward, being adequate for closed-loop systems. Additionally, the constraint equations necessary to describe the system restrictions are often simple to obtain. Furthermore, this approach exhibits good computational efficiency and the degree of nonlinearity of the resulting equations is low. With absolute coordinates, the configurations of the systems are defined in a univocal manner. The major drawback associated with the absolute coordinates formulation is the large number of variables and constraint equations involved (Nikravesh 1988, Shabana 1989, Jálon and Bayo 1994, Nikravesh 2008).

As far as the relative coordinates is concerned, it can be said that this type of coordinates was primarily used in the development of the first general computer programs for mechanisms analysis (Paul and Krajcinovic 1970, Sheth and Uicker 1971). The relative coordinates, also denominated as joint coordinates or state variables, define the position and orientation of a body with respect to a preceding body in a multibody system. In general, this type of coordinates is directly associated with the relative degrees of freedom allowed by joints that connect bodies. Relative coordinates can be associated with linear or angular displacements, as Figs. 23 and 24 demonstrate.

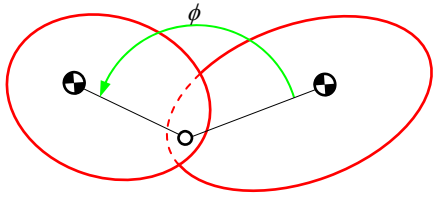


Fig. 23 Relative coordinate: angular displacement

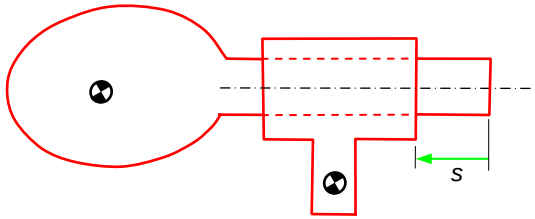


Fig. 24 Relative coordinate: linear displacement

Relative Coordinates

Considering the four-bar mechanism of Fig. 25, the set of relative coordinates that define its configuration is

$$\mathbf{q} = \{\phi_1 \quad \phi_2 \quad \phi_3\}^T \tag{15}$$

in which the three variables ϕ_1, ϕ_2 and ϕ_3 represent the angle of each body with respect to the *x*-axis. Since the four-bar mechanism has only one degree of freedom, then the three relative coordinates ϕ_1, ϕ_2 and ϕ_3 are not independent, being necessary to write a set of two constraint equations. In general, when working with relative coordinates, these equations can be obtained from the closed kinematic chain, yielding

$$a\cos\phi_1 + b\cos\phi_2 + c\cos\phi_3 - d = 0 \tag{16}$$

$$a\sin\phi_1 + b\sin\phi_2 + c\sin\phi_3 = 0 \tag{17}$$

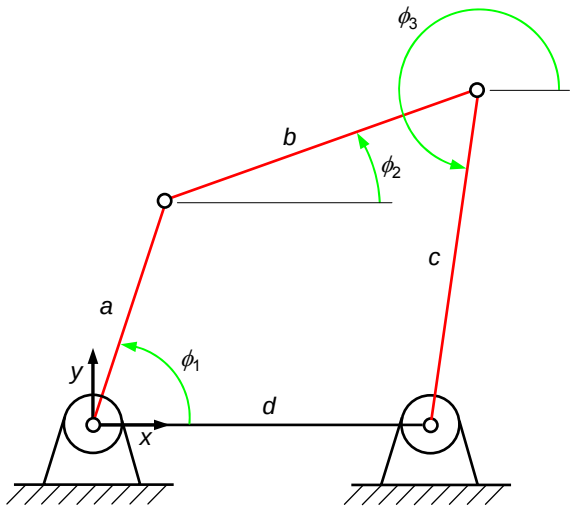


Fig. 25 Four-bar mechanism described by relative coordinates

Relative Coordinates

Thus, for a given configuration, that is, known for instance ϕ_1, the set algebraic equations (16) and (17) must be solved simultaneously for ϕ_2 and ϕ_3. This procedure is generally preformed numerically.

In summary, the relative coordinates are used to formulate a minimum number of equations of motion of multibody systems.

When the system is an open kinematic chain, the number of relative coordinates is equal to the number of degrees of freedom, as it is the example of the triple pendulum illustrated in Fig. 19.

In these circumstances, the relative coordinates are in fact the independent variables used to define the configuration of the system.

For closed kinematic chains a preprocessing analysis of the system is required to deal with the assembling constraints, and then the system topology has to be analyzed to find how to write them properly.

Therefore, relative coordinates are not convenient when the system topology can be altered during the global motion produced.

Furthermore, in sharp contrast to the absolute coordinates, the incorporation of general force functions, constraint equations and prescribed trajectories in the system's formulation is not trivial task when relative coordinates are used.

Another disadvantage associated with this type of coordinates is that they generate equations of high level of nonlinearity and, hence, their computational implementation is hard. Nevertheless, the use of relative coordinates presents a good computational efficiency and defines the system in a univocal manner.

Natural Coordinates

Natural coordinates, also designated as point coordinates or fully Cartesian coordinates, are an interesting alternative to absolute or relative coordinates when describing multibody systems. Jálon (2007), who is the father of the natural coordinates, developed the concept of natural coordinates in the early eighties based on the matrix analysis of structures.

In a simple manner, two main ingredients characterize the natural coordinates, that is, they are composed by the Cartesian coordinates of some points and by the Cartesian coordinates of some unit vectors distributed on the different bodies of the system.

The points are typically located in relevant positions of the multibody system components, such as joints and extremities of the bodies. In turn, the vectors are normally used to define rotational and direction axes for kinematic joints. The natural coordinates have the great advantage that there is no need for any angular variables (Jálon and Bayo 1994), as it is the case of absolute and relative coordinates.

In two-dimensional space, the natural coordinates can be seen as an extension of the absolute coordinates when the reference points are moved to relevant points of the multibody system. Figure 26 illustrates a schematic representation of this basic idea of the transition from absolute coordinates to natural coordinates.

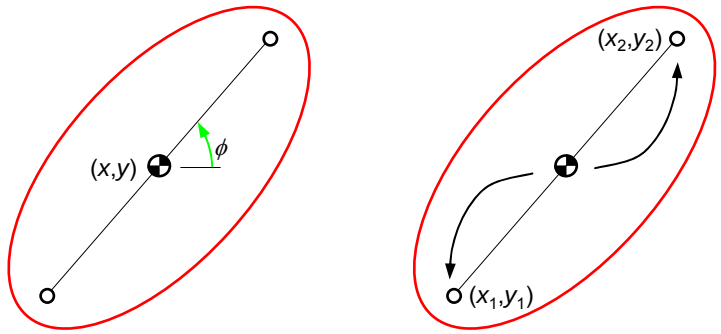


Fig. 26 Transition from absolute to natural coordinates

Natural Coordinates

For instance, in the case of the four-bar mechanism depicted in Fig. 27 the system can be described by four Cartesian coordinates x_2, y_2, x_3 and y_3, that is

$$\mathbf{q} = \{x_2 \quad y_2 \quad x_3 \quad y_3\}^T \tag{18}$$

Since the four-bar mechanism has only one degree of freedom, then three constraint equations are required to describe the configuration of the system. The constraint equations must ensure that points 2 and 3 move according to the restrictions imposed on them by the three moving rigid bodies. Thus, based on the concept of rigid bodes (defined by constant distances between points) the three following conditions can be written

$$(x_2 - x_1)^2 + (y_2 - y_1)^2 - a^2 = 0 \tag{19}$$

$$(x_3 - x_2)^2 + (y_3 - y_2)^2 - b^2 = 0 \tag{20}$$

$$(x_3 - x_4)^2 + (y_3 - y_4)^2 - c^2 = 0 \tag{21}$$

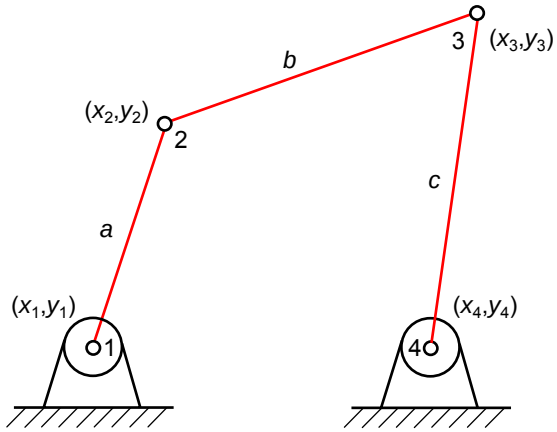


Fig. 27 Four-bar mechanism described by natural coordinates

Natural Coordinates

By analyzing Eqs. (19), (20) and (21) it can be observed that no angular variables are involved and they are nonlinear equations, namely, quadratic in this particular example of the four-bar mechanism.

Natural coordinates are especially appropriate to sensitive analysis and optimization procedures, because the lengths, for instance, appear explicitly in the constraint equations. This is not the case of absolute and relative coordinates.

Furthermore, when using natural coordinates there is no need for preprocessing, as in the case of the closed kinematic chain systems described by relative coordinates. The natural coordinates are not independent but they are related by the rigid body condition, that is, the condition of keeping constant distances and angles.

Thus, the constraint equations can be formulated with the scalar product vectors, leading to quadratic constraint equations and linear terms in the Jacobian matrix (Jálon 2007).

In conclusion, there are different manners of describing the configuration of a multibody system. In other words, there are many types of coordinates that can be helpful in the formulation of the equations of motion for multibody systems.

The dilemma of selection of the type of coordinates to be used depends on the type of problem to be analyzed. In fact, the choice of the most appropriate set of coordinates is not indifferent, being a trade-off between the advantages and drawbacks associated with each type of coordinates.

A valuable comparison of the main types of coordinates are presented and discussed by Nikravesh (1988), Shabana (1989) and Jálon and Bayo (1994), where the plus and minus of each type of coordinates are highlighted.

Shabana (1989) called attention for the selection of the coordinates in flexible multibody systems, which is a much more difficult task.

Global and Local Coordinates

Displacements, velocities and accelerations are quantities frequently used to characterize the configuration and motion properties of the multibody systems.

For this purpose, a proper system of coordinates must be adopted, which includes the global and local systems of coordinates.

The expression global coordinate system, which is represented by two orthogonal axes that are rigidly connected at a point called origin of this system, is used to represent the inertial frame of reference. In the present work, the global coordinate system is denoted by *xy*.

In addition, a body-fixed or local coordinate system is considered to define local properties of points that belong to a body *i*. This local system of coordinates is, in general, attached to the center of mass of the bodies and is denoted by $\xi_i \eta_i$. This local system translates and rotates with the body motion, consequently, its location and rotation vary with time.

Figure 28 shows the global and local systems of coordinates in a multibody system composed by n_b rigid bodies (Flores et al. 2008).

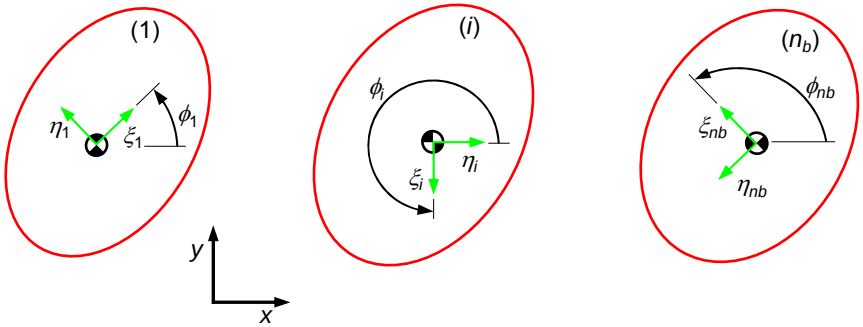


Fig. 28 Global and local systems of coordinates

Global and Local Coordinates

If a multibody system is made of n_b rigid bodies such as the one illustrated in Fig. 28, then the number of absolute coordinates is $n=3\times n_b$. Thus, the vector of generalized coordinates of this system can be written as

$$\mathbf{q} = \left\{ \mathbf{q}_1^T \quad \mathbf{q}_2^T \quad \cdots \quad \mathbf{q}_{n_b}^T \right\}^T \tag{22}$$

Let now consider a single body, denoted as body i, that is part of a multibody system, as Fig. 29 depicted. When absolute coordinates are used, the position and orientation of the body are defined by a set of translational and rotational coordinates. Thus, body i is uniquely located in the plane by specifying the global position, $\mathbf{r}_i$, of the body-fixed coordinate system origin, O_i, and the angle ϕ_i of rotation of this system of coordinates with respect to the x-axis of the global coordinate system. Finally, the vector of coordinates of the body i is denoted by

$$\mathbf{q}_i = \left\{ x_i \quad y_i \quad \phi_i \right\}^T \tag{23}$$

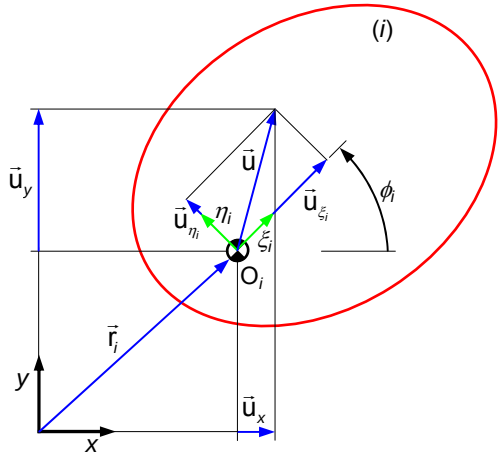


Fig. 29 Unit vector components in global and local systems of coordinates

Global and Local Coordinates

Let $\mathbf{u}_x$ and $\mathbf{u}_y$ be unit vectors along the global x and y axes, respectively, and let $\mathbf{u}_{\xi i}$ and $\mathbf{u}_{\eta i}$ be unit vectors along the body-fixed axes ξ_i and η_i, respectively. It is known that the unit vectors $\mathbf{u}_x$ and $\mathbf{u}_y$ are fixed in time, that is, they have constant magnitude and direction. In turn, the unit vectors $\mathbf{u}_{\xi i}$ and $\mathbf{u}_{\eta i}$ have constant magnitude but have changeable orientations. The unit vector components $\mathbf{u}_x$ and $\mathbf{u}_y$ of vector $\mathbf{u}$ is written as

$$\mathbf{u}_x = \mathbf{u}_{\xi_i} \cos\phi_i - \mathbf{u}_{\eta_i} \sin\phi_i \tag{24}$$

$$\mathbf{u}_y = \mathbf{u}_{\xi_i} \sin\phi_i + \mathbf{u}_{\eta_i} \cos\phi_i \tag{25}$$

In matrix form, Eqs. (24) and (25) can be expressed as

$$\begin{Bmatrix} \mathbf{u}_x \\ \mathbf{u}_y \end{Bmatrix} = \begin{bmatrix} \cos\phi_i & -\sin\phi_i \\ \sin\phi_i & \cos\phi_i \end{bmatrix} \begin{Bmatrix} \mathbf{u}_{\xi_i} \\ \mathbf{u}_{\eta_i} \end{Bmatrix} \tag{26}$$

and in a compact form results in

$$\mathbf{u} = \mathbf{A}_i \mathbf{u}'_i \tag{27}$$

where $\mathbf{u}$ is the unit vector expressed in terms of global coordinates, $\mathbf{u}'_i$ is the unit vector expressed in the local coordinate system and $\mathbf{A}_i$ represents the planar transformation matrix for body i, which defines the orientation of body-fixed coordinate system $\xi_i \eta_i$ with respect to the global coordinate system xy, being given by

$$\mathbf{A}_i = \begin{bmatrix} \cos\phi_i & -\sin\phi_i \\ \sin\phi_i & \cos\phi_i \end{bmatrix} \tag{28}$$

Global and Local Coordinates

In the present work, the body-fixed or local components of a vector are denoted by an apostrophe, such as $\mathbf{u}'_i$. The analysis presented above is only valid for rigid bodies, that is, bodies in which the distances among their particles do not change during the motion of the body. Therefore, each point in a rigid body is located by its constant position vector expressed in the body-fixed coordinate system. For instance, a point P_i on body i can be described by the position vector $\mathbf{s}_i^P$ and by the global position of the body center of mass $\mathbf{r}_i$, resulting that

$$\mathbf{r}_i^P = \mathbf{r}_i + \mathbf{s}_i^P = \mathbf{r}_i + \mathbf{A}_i \mathbf{s}'^P_i \tag{29}$$

where $\mathbf{A}_i$ is the transformation matrix given by Eq. (28) and $\mathbf{s}'^P_i$ refers to the local components of point P_i.

The location of point P_i with respect to body-fixed coordinate system is

$$\mathbf{s}'^P_i = \left\{ \xi_i^P \quad \eta_i^P \right\}^T \tag{30}$$

In the expanded form, Eq. (29) is expressed as

$$\begin{Bmatrix} x_i^P \\ y_i^P \end{Bmatrix} = \begin{Bmatrix} x_i \\ y_i \end{Bmatrix} + \begin{bmatrix} \cos\phi_i & -\sin\phi_i \\ \sin\phi_i & \cos\phi_i \end{bmatrix} \begin{Bmatrix} \xi_i^P \\ \eta_i^P \end{Bmatrix} \tag{31}$$

or, alternatively,

$$x_i^P = x_i + \xi_i^P \cos\phi_i - \eta_i^P \sin\phi_i \tag{32}$$

$$y_i^P = y_i + \xi_i^P \sin\phi_i + \eta_i^P \cos\phi_i \tag{33}$$

Global and Local Coordinates

Figure 30 illustrates the global and local components of a point P_i on body i.

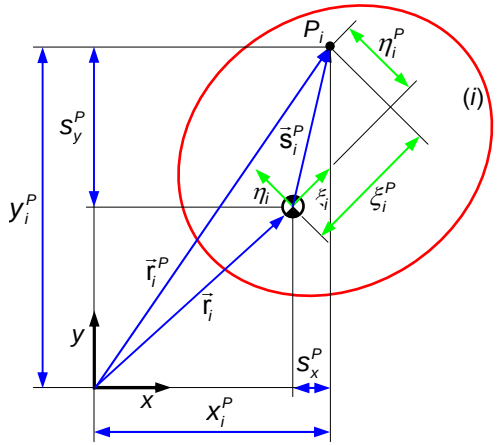


Fig. 30 Global and local components of a point P_i on body i

It should be noted that

$$\mathbf{s}_i^P = \mathbf{A}_i \mathbf{s}'^P_i \tag{34}$$

is the relation between local and global components of point P_i. Equations (23) through (34) form some of the fundamental mathematical relations in the study of the motion characteristics of multibody systems.

The velocity components of point P_i can be obtained by simple differentiate Eqs. (32) and (33) with respect to time, yielding

$$\dot{x}_i^P = \dot{x}_i - (\xi_i^P \sin\phi_i + \eta_i^P \cos\phi_i)\dot{\phi}_i \tag{35}$$

$$\dot{y}_i^P = \dot{y}_i + (\xi_i^P \cos\phi_i - \eta_i^P \sin\phi_i)\dot{\phi}_i \tag{36}$$

Global and Local Coordinates

A second differentiation of Eqs. (32) and (33) with respect to time leads to acceleration equations, obtained as

$$\ddot{x}_i^P = \ddot{x}_i - (\xi_i^P \sin\phi_i + \eta_i^P \cos\phi_i)\ddot{\phi}_i - (\xi_i^P \cos\phi_i - \eta_i^P \sin\phi_i)\dot{\phi}_i^2 \tag{37}$$

$$\ddot{y}_i^P = \ddot{y}_i + (\xi_i^P \cos\phi_i - \eta_i^P \sin\phi_i)\ddot{\phi}_i - (\xi_i^P \sin\phi_i + \eta_i^P \cos\phi_i)\dot{\phi}_i^2 \tag{38}$$

Alternatively, the velocity of a point P_i on body i in the global coordinate system can be obtained by taking the time derivative of Eq. (29)

$$\dot{\mathbf{r}}_i^P = \dot{\mathbf{r}}_i + \dot{\mathbf{A}}_i \mathbf{s}'^P_i = \dot{\mathbf{r}}_i + \mathbf{B}_i \mathbf{s}'^P_i \dot{\phi}_i \tag{39}$$

in which

$$\mathbf{B}_i = \begin{bmatrix} -\sin\phi_i & -\cos\phi_i \\ \cos\phi_i & -\sin\phi_i \end{bmatrix} \tag{40}$$

In a similar way, the time derivative of Eq. (39) yields the acceleration of the point P_i, that is

$$\ddot{\mathbf{r}}_i^P = \ddot{\mathbf{r}}_i + \ddot{\mathbf{A}}_i \mathbf{s}'^P_i = \ddot{\mathbf{r}}_i + \mathbf{B}_i \mathbf{s}'^P_i \ddot{\phi}_i - \mathbf{A}_i \mathbf{s}'^P_i \dot{\phi}_i^2 \tag{41}$$

Generalities and Degrees of Freedom

The main purpose of this section is to present the formulation of the most common types of kinematic constraint equations used to model multibody systems. It is also objective to introduce their contributions to the Jacobian matrix of the constraints and to the right-hand side of the velocity and acceleration constraint equations. These terms form the basis to perform the kinematic analysis of multibody systems. In turn, the kinematic analysis is of a preliminary nature and plays an important role in the understanding the dynamics of moving bodies within a multibody system.

It is known that the degrees of freedom in a multibody system are directly related to the types of kinematic constraints considered, namely, those associated with kinematic joints. Furthermore, each type of joint allows for certain relative motions between adjacent bodies and constrains others. Figure 31 shows the basic kinematic lower joints most frequently used when modeling, simulating and analyze multibody systems, in which their denomination, geometric representation and relative degrees of freedom permitted are presented (Reuleaux 1963, Nikravesh 1988, Shabana 1989, Haug, 1989, Flores and Claro 2007b).

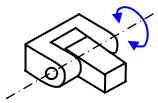
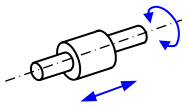
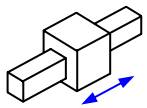
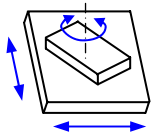
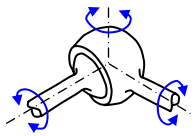
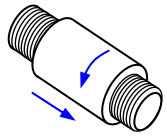
Denomination	Geometric representation	Degrees of freedom	Denomination	Geometric representation	Degrees of freedom
Revolute		1	Cylindrical		2
Translational or Prismatic		1	Planar		3
Spherical or globular		3	Screw		1

Fig. 31 Basic kinematic lower joints frequently used in multibody dynamics

Generalized Coordinates and Constraints

As it was presented in the previous section, the configuration of a multibody system is described by a set of variables called generalized variables that completely define the location and orientation of each body in the system.

Hereafter, the set of generalized coordinates of a multibody system will be denoted be vector **q**, which can be written as $\mathbf{q}=\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \ldots, \mathbf{q}_n\}^T$, where *n* is the number of coordinates.

In multibody systems formulation with absolute coordinates, the generalized coordinates can be divided into independent and dependent variables, hence, several algebraic equations are needed to be introduced to relate them.

In other words, the constraint equations represent the kinematic relation between independent and dependent coordinates.

In a simple manner, the constraint equations can arise from the description of the system topology and from the characterization of the driving and guiding constraints that are used to guide the system through the analysis.

In the present text, the set of constraint equations is denoted by symbol Φ.

In order to distinguish among the different constraint equations, each elementary set of constraints is identified by a superscript containing two parameters.

The first parameter denotes the type of constraint, while the second one defines the number of independent equations that it involves.

For example, $\Phi^{(r,2)}$ refers to a revolute (*r*) joint constraint, which contains two (2) equations.

Kinematic Constraints

Kinematic constraints can be classified as holonomic or nonholonomic.

Holonomic constraints arise from geometric constraints and are integrable into a form involving only coordinates (*holo* comes from Greek that means whole, integer). Nonholonomic constraints are not integrable.

The relation specified by a constraint can be an explicit function of time designated as rheonomic constraints (*rheo* comes from Greek that means hard, inflexible, independent) or not, being designated by scleronomic constraints (*scleros* comes from Greek that means flexible, changing).

Figure 32 shows a typical revolute joint and a simple human body model placed on a spherical surface, which represents a holonomic and a nonholonomic constraint, respectively. Thus, for instance, in the motion of the human model on the spherical surface, the following mathematical relation has to be satisfied during the analysis (Flores 2006a)

$$\mathbf{r}^T\mathbf{r} - R^2 \geq \mathbf{0} \tag{42}$$

where R is the radius of the spherical surface and vector $\mathbf{r}$ represents the position of the model measured from the center of the spherical surface.

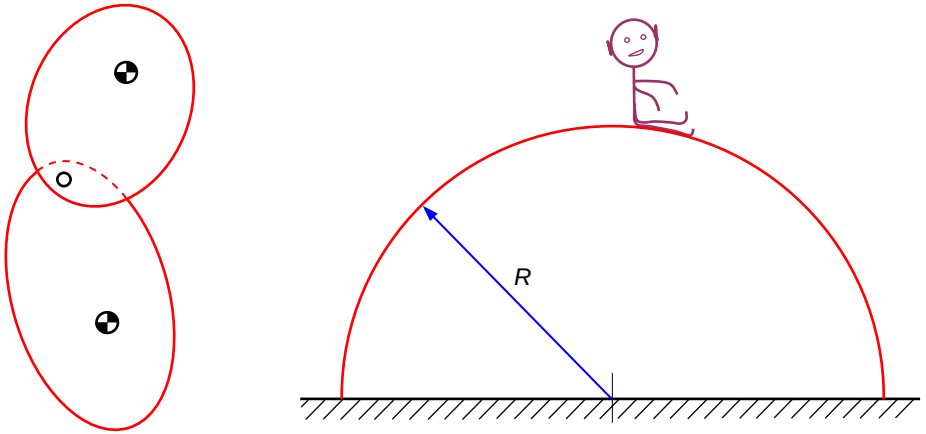


Fig. 32 Holonomic constraint and nonholonomic constraint

Kinematic Constraints and Kinematic Analysis

The kinematic constraints considered here are assumed to be holonomic, arising from geometrical constraints on the generalized coordinates.

Holonomic constraints, also called geometric restrictions, are algebraic equations imposed to the system that are expressed as functions of the displacement and, possibly, time.

If the time *t* does not appear explicitly in the constraint equation, then the system is said to be scleronomic.

A simple example of scleronomic constraint equation is the revolute joint between two bodies.

Otherwise, when the constraint is holonomic and *t* appears explicitly, the system is said to be rheonomic.

When the constraint equation contains inequalities or relations between velocity components that are not integrable in closed form, they are denominated as nonholonomic.

Nonholonomic constraints are still kinematic constraints since they impose restrictions to the velocity and acceleration (Nikravesh 1988, Shabana 1989).

Let the configuration of a multibody system be described by *n* absolute coordinates, then a set of *m* algebraic kinematic independent holonomic constraints Φ can be written in a compact form as (Nikravesh 1988).

$$\boldsymbol{\Phi}(\mathbf{q}, t) = \mathbf{0} \tag{43}$$

where **q** is the vector of generalized coordinates and *t* is the time variable, in general associated with the driving and guiding elements.

Kinematic Analysis

The velocities and accelerations of the system elements are evaluated using the velocity and acceleration constraint equations. Consequently, the first time derivative of Eq. (43) provides the velocity constraint equations as

$$\mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} = -\mathbf{\Phi}_{t} = \boldsymbol{\upsilon} \tag{44}$$

where $\mathbf{\Phi_q}$ is the Jacobian matrix of the constraint equations, that is, the matrix of the partial derivatives, $\partial\mathbf{\Phi}/\partial\mathbf{q}$, is the vector of generalized velocities and $\boldsymbol{\upsilon}$ is the right-hand side of velocity constraint equations, which contains the partial derivatives of $\mathbf{\Phi}$ with respect to time, $\partial\mathbf{\Phi}/\partial t$.

Notice that only rheonomic constraints, associated with driving or guiding equations, contribute with non-zero entries to the vector $\boldsymbol{\upsilon}$ (Flores 2006b).

A second differentiation of Eq. (43) with respect to time leads to the acceleration constraint equations, yielding

$$\mathbf{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}} = -(\mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}})_{\mathbf{q}}\dot{\mathbf{q}} - 2\mathbf{\Phi}_{\mathbf{q}t}\dot{\mathbf{q}} - \mathbf{\Phi}_{tt} = \boldsymbol{\gamma} \tag{45}$$

where $\ddot{\mathbf{q}}$ is the vector of generalized acceleration and $\boldsymbol{\gamma}$ is the right-hand side of acceleration constraint equations, that is, the vector of quadratic velocity terms, which are exclusively functions of velocity, position and time.

In the case of scleronomic constraints, that is, when $\mathbf{\Phi}$ is not explicitly dependent on the time, the terms $\mathbf{\Phi}_t$ in Eq. (44) and $\mathbf{\Phi}_{\mathbf{q}t}$ and $\mathbf{\Phi}_{tt}$ in Eq. (45) vanish.

Kinematic Analysis

In should be highlighted that the terms involved in Eqs. (43) through (45) appear in a general form, that is, they do not reflect the type of coordinates considered.

In addition, the constraint equations represented by Eq. (43) are non linear in terms of **q** and are, usually, solved by employing the Newton-Raphson method.

Equations (44) and (45) are linear in terms of $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$, respectively, and can be solved by any usual method adopted for the solution of systems of linear equations.

It should be noted that the issues related to the treatment of redundant constraints are not presented in this work.

The interested reader in the details on this particular topic is referred to the work by Wehage and Haug (1982).

The kinematic analysis is the study of the motion characteristics of a multibody system, independently of the causes that produce it.

Since in the kinematic analysis the forces are not considered, the motion of the system is specified by driving or guiding elements that govern the motion of specific degrees of freedom of the system during the analysis.

The position, velocity and acceleration of the remaining elements of the system are defined by kinematic constraint equations that describe the system topology.

It is clear that in the kinematic analysis, the number of driving and guiding constraints must be equal to the number of degrees of freedom of the multibody system.

In short, the kinematic analysis is performed by solving a set of equations that result from the kinematic, driving and guiding constraints.

Kinematic Analysis

The kinematic analysis of a multibody system can be carried out by solving the set of Eqs. (43)-(45). The necessary steps to perform this type of analysis, sketched in Fig. 33, are summarized as

1. Specify initial conditions for positions $\mathbf{q}^0$ and initialize the time t_0.
2. Evaluate the position constraint equations (43) and solve them for positions, $\mathbf{q}$.
3. Evaluate the velocity constraint equations (44) and solve them for velocities, $\dot{\mathbf{q}}$.
4. Evaluate the acceleration constraint equations (45) and solve them for accelerations, $\ddot{\mathbf{q}}$.
5. Increment the time. If the time is smaller than final time, go to step 2), otherwise stop the analysis.

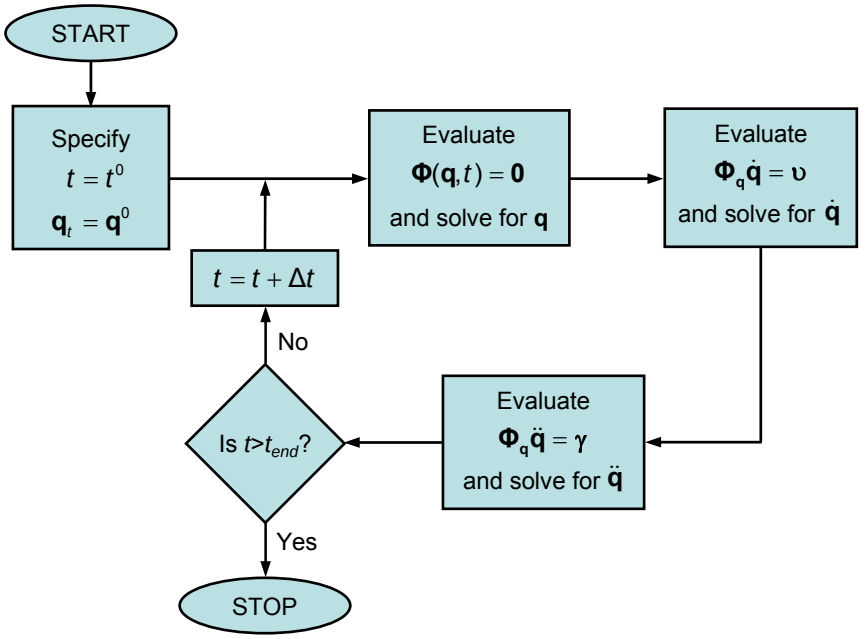


Fig. 33 Flowchart of computational procedure for kinematic analysis of a multibody system

Constraint Equations

A close observation of the Eqs. (44) and (45) shows that both expressions represent systems of linear equations, with the same leading matrix and different right-hand side vectors. Moreover, since both expressions share the same leading matrix, Jacobian matrix of the constraints, evaluated with the latest calculated configuration of the system, then this matrix only needs to be factorized once during each step (Nikravesh 1988).

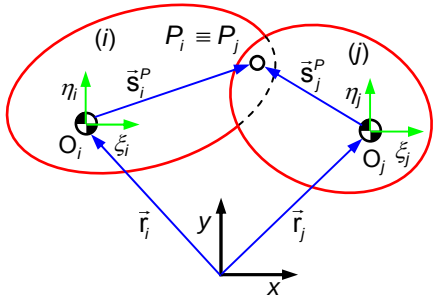


Fig. 34 Revolute joint connecting bodies *i* and *j*

Figure 34 shows two bodies connected by a revolute joint, which is a pin and bush type of joint that constrains the relative translation between the two bodies *i* and *j*, allowing only the relative rotation.

The kinematic conditions for the revolute joint require that two different points, each one belonging to a different body, share the same position in space all the time. This means that the global position of the point P_i on body *i* is coincident with the global position of the point P_j on body *j*.

Such condition is expressed by two algebraic equations that can be obtained from the following vector loop equation.

$$\mathbf{r}_i + \mathbf{s}_i^P - \mathbf{r}_j - \mathbf{s}_j^P = \mathbf{0} \tag{46}$$

Constraint Equations

which can be re-written as

$$\mathbf{\Phi}^{(r,2)} \equiv \mathbf{r}_i + \mathbf{A}_i \mathbf{s}'^P_i - \mathbf{r}_j - \mathbf{A}_j \mathbf{s}'^P_j = \mathbf{0} \tag{47}$$

In a more explicit form, Eq. (47) can be expressed by

$$\mathbf{\Phi}^{(r,2)} \equiv \begin{Bmatrix} x_i^P - x_j^P \\ y_i^P - y_j^P \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{48}$$

Expanding Eq. (48) yields two constraint equations as

$$\mathbf{\Phi}^{(r,1st)} \equiv x_i + \xi_i^P \cos\phi_i - \eta_i^P \sin\phi_i - x_j - \xi_j^P \cos\phi_j + \eta_j^P \sin\phi_j = 0 \tag{49}$$

$$\mathbf{\Phi}^{(r,2nd)} \equiv y_i + \xi_i^P \sin\phi_i + \eta_i^P \cos\phi_i - y_j - \xi_j^P \sin\phi_j - \eta_j^P \cos\phi_j = 0 \tag{50}$$

In summation, there is only one relative degree of freedom between two bodies that are connected by a revolute joint.

In other words, the two constraint equations (49) and (50) reduce the number of degrees of freedom of the system by two.

Constraint Equations

In order to perform the kinematic analysis, it is necessary to evaluate the Jacobian matrix of the constraint equations for positions and the right-hand side of the velocity and acceleration constraint equations.

Thus, the Jacobian matrix associated with the revolute joint is, by definition, the partial derivatives of Eqs. (49) and (50) with respect to generalized coordinates **q**, which is a column vector written as

$$\mathbf{q} = \begin{Bmatrix} x_i & y_i & \phi_i & x_j & y_j & \phi_j \end{Bmatrix}^T \tag{51}$$

Hence, the Jacobian matrix is given by

$$\boldsymbol{\Phi}_{\mathbf{q}}^{(r,2)} = \begin{bmatrix} \dfrac{\partial \boldsymbol{\Phi}^{(r,1st)}}{\partial x_i} & \dfrac{\partial \boldsymbol{\Phi}^{(r,1st)}}{\partial y_i} & \dfrac{\partial \boldsymbol{\Phi}^{(r,1st)}}{\partial \phi_i} & \dfrac{\partial \boldsymbol{\Phi}^{(r,1st)}}{\partial x_j} & \dfrac{\partial \boldsymbol{\Phi}^{(r,1st)}}{\partial y_j} & \dfrac{\partial \boldsymbol{\Phi}^{(r,1st)}}{\partial \phi_j} \\ \dfrac{\partial \boldsymbol{\Phi}^{(r,2nd)}}{\partial x_i} & \dfrac{\partial \boldsymbol{\Phi}^{(r,2nd)}}{\partial y_i} & \dfrac{\partial \boldsymbol{\Phi}^{(r,2nd)}}{\partial \phi_i} & \dfrac{\partial \boldsymbol{\Phi}^{(r,2nd)}}{\partial x_j} & \dfrac{\partial \boldsymbol{\Phi}^{(r,2nd)}}{\partial y_j} & \dfrac{\partial \boldsymbol{\Phi}^{(r,2nd)}}{\partial \phi_j} \end{bmatrix} \tag{52}$$

The evaluation of the partial derivatives that appear in Eq. (52) yield the corresponding values of the Jacobian terms as follows

$$\boldsymbol{\Phi}_{\mathbf{q}}^{(r,2)} = \begin{bmatrix} 1 & 0 & -\xi_i^P \sin\phi_i - \eta_i^P \cos\phi_i & -1 & 0 & \xi_i^P \sin\phi_i + \eta_i^P \cos\phi_i \\ 0 & 1 & \xi_i^P \cos\phi_i - \eta_i^P \sin\phi_i & 0 & -1 & -\xi_i^P \cos\phi_i + \eta_i^P \sin\phi_i \end{bmatrix} \tag{53}$$

Constraint Equations

Alternatively, Eq. (53) can be written in an explicit form as

$$\mathbf{\Phi}_{\mathbf{q}}^{(r,2)} = \begin{bmatrix} 1 & 0 & -y_i^P + y_i & -1 & 0 & y_j^P - y_j \\ 0 & 1 & x_i^P - x_i & 0 & -1 & -x_j^P + x_j \end{bmatrix} \tag{54}$$

In turn, the right-hand side of the velocity constraint equations is

$$\boldsymbol{\upsilon}^{(r,2)} = -\mathbf{\Phi}_t = \mathbf{0} \tag{55}$$

or, alternatively

$$\boldsymbol{\upsilon}^{(r,2)} = -\mathbf{\Phi}_t = \begin{Bmatrix} \dfrac{\partial \mathbf{\Phi}^{(r,1\text{st})}}{\partial t} \\ \dfrac{\partial \mathbf{\Phi}^{(r,2\text{nd})}}{\partial t} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{56}$$

Equation (55) represents a scleronomic constraint, that is, the revolute joint constraints do not depend explicitly on time. Consequently, the right-hand side of the acceleration constraint equations reduces to

$$\boldsymbol{\gamma}^{(r,2)} = -(\mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}})_{\mathbf{q}}\,\dot{\mathbf{q}} \tag{57}$$

Constraint Equations

which results in

$$\boldsymbol{\gamma}^{(r,2)} = \begin{bmatrix} (\xi_i^P \cos\phi_i - \eta_i^P \sin\phi_i)\dot{\phi}_i^2 - (\xi_j^P \cos\phi_j - \eta_j^P \sin\phi_j)\dot{\phi}_j^2 \\ (\xi_i^P \sin\phi_i + \eta_i^P \cos\phi_i)\dot{\phi}_i^2 - (\xi_j^P \sin\phi_j + \eta_j^P \cos\phi_j)\dot{\phi}_j^2 \end{bmatrix}$$
$$= \begin{bmatrix} (x_i^P - x_i)\dot{\phi}_i^2 - (x_j^P - x_j)\dot{\phi}_j^2 \\ (y_i^P - y_i)\dot{\phi}_i^2 - (y_j^P - y_j)\dot{\phi}_j^2 \end{bmatrix} \tag{58}$$
$$= \mathbf{s}_i^P \dot{\phi}_i^2 - \mathbf{s}_j^P \dot{\phi}_j^2 = \mathbf{A}_i \mathbf{s}'^P_i \dot{\phi}_i^2 - \mathbf{A}_j \mathbf{s}'^P_j \dot{\phi}_j^2$$

An alternative manner to obtain the Jacobian matrix of the constraints and the right-hand side of the velocity constraint equations is to derivate the constraint equations (49) and (50) with respect to time, yielding

$$\dot{x}_i - (\xi_i^P \sin\phi_i + \eta_i^P \cos\phi_i)\dot{\phi}_i - \dot{x}_j + (\xi_j^P \sin\phi_j + \eta_j^P \cos\phi_j)\dot{\phi}_j = 0 \tag{59}$$

$$\dot{y}_i + (\xi_i^P \cos\phi_i - \eta_i^P \sin\phi_i)\dot{\phi}_i - \dot{y}_j - (\xi_j^P \cos\phi_j - \eta_j^P \sin\phi_j)\dot{\phi}_j = 0 \tag{60}$$

These two equations can be written in matrix form as

$$\boldsymbol{\Phi}_{\mathbf{q}}^{(r,2)} \left\{ \dot{x}_i \quad \dot{y}_i \quad \dot{\phi}_i \quad \dot{x}_j \quad \dot{y}_j \quad \dot{\phi}_j \right\}^T = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{61}$$

which the Jacobian matrix is given by Eq. (53). Equation (61) represents the velocity constraint equations for a revolute joint. It should be highlighted that the right-hand side, υ, which represents the time derivative of the constraints, is null because a revolute joint is of scleronomic type.

Constraint Equations

Similarly, the acceleration constraint equations of a revolute joint can be obtained by taking the time derivative of Eqs. (59) and (60), yielding

$$\ddot{x}_i - (\xi_i^P \sin\phi_i + \eta_i^P \cos\phi_i)\ddot{\phi}_i - (\xi_i^P \cos\phi_i - \eta_i^P \sin\phi_i)\dot{\phi}_i^2 - \ddot{x}_j + (\xi_j^P \sin\phi_j + \eta_j^P \cos\phi_j)\ddot{\phi}_j + (\xi_j^P \cos\phi_j - \eta_j^P \sin\phi_j)\dot{\phi}_j^2 = 0 \qquad (62)$$

$$\ddot{y}_i + (\xi_i^P \cos\phi_i - \eta_i^P \sin\phi_i)\ddot{\phi}_i - (\xi_i^P \sin\phi_i + \eta_i^P \cos\phi_i)\dot{\phi}_i^2 - \ddot{y}_j - (\xi_j^P \cos\phi_j - \eta_j^P \sin\phi_j)\ddot{\phi}_j + (\xi_j^P \sin\phi_j + \eta_j^P \cos\phi_j)\dot{\phi}_j^2 = 0 \qquad (63)$$

or in a compact form

$$\mathbf{\Phi}_{\mathbf{q}}^{(r,2)} \begin{Bmatrix} \ddot{x}_i \\ \ddot{y}_i \\ \ddot{\phi}_i \\ \ddot{x}_j \\ \ddot{y}_j \\ \ddot{\phi}_j \end{Bmatrix} = \boldsymbol{\gamma}^{(r,2)} \qquad (64)$$

where the vector $\gamma^{(r,2)}$ is given by Eq. (58).

Constraint Equations

In what follows, the main kinematic variables associated with the translational joints are briefly presented. The interested reader in the detailed formulation is referred to the works by Nikravesh (1988) and Haug (1989).

Figure 35 shows two bodies *i* and *j* connected by a translational joint, in which the slider and guide can translate with respect to each other parallel to the line of translation.

So, there is neither rotation between the bodies nor a relative translation motion in the direction perpendicular to the line of translation.

Thus, a translational joint reduces the number of degrees of freedom of the system by two, which implies the need for two independent algebraic equations to represent it.

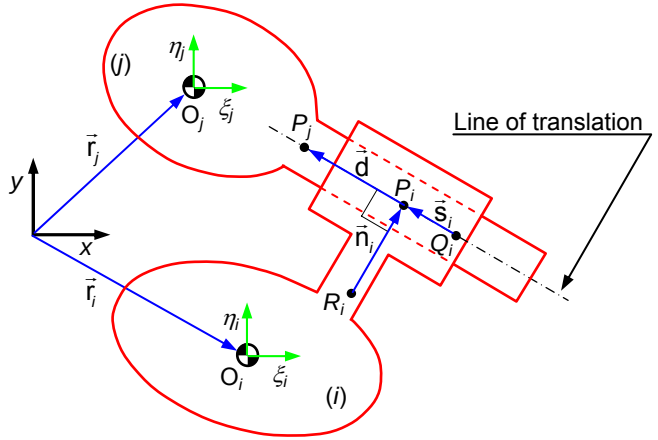


Fig. 35 Translational joint connecting bodies *i* and *j*

A constraint equation for eliminating the relative rotation between the two bodies *i* and *j* is written as

$$\phi_i - \phi_j - (\phi_i^0 - \phi_j^0) = 0 \tag{65}$$

where ϕ_i^0 and ϕ_j^0 are the initial rotational angles for each body.

Constraint Equations

In order to eliminate the relative translation motion between the two bodies in a direction perpendicular to the line of translation, the two vectors $\mathbf{s}_i$ and $\mathbf{d}$ shown in Fig. 35 must remain parallel during the motion of the system.

These vectors are defined by locating three points on the line of translation, two points on body *i* and one point on body *j*. This condition is imposed by forcing the vector product of these two vectors to remain null all the time.

A simpler and alternative method consists of defining another vector $\mathbf{n}_i$ fixed to body *i* and perpendicular to the line of translation. Then, it is only required that vector $\mathbf{d}$ remains perpendicular to vector $\mathbf{n}_i$, that is

$$\mathbf{n}_i^T \mathbf{d} = 0 \tag{66}$$

Therefore, Eqs. (66) and (65) yield the two necessary constraint equations for a translational joint, that can be written in an explicit form as

$$\mathbf{\Phi}^{(t,2)} = \begin{Bmatrix} (x_i^P - x_i^Q)(y_j^P - y_i^P) - (y_i^P - y_i^Q)(x_j^P - x_i^P) \\ \phi_i - \phi_j - (\phi_i^0 - \phi_j^0) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{67}$$

where x_i^P and y_i^P are the global coordinates of point P_i on body *i*, and the coordinates of points Q_i on body *i* and P_j on body *j* follow the same notation.

Constraint Equations

Using the same approach as in the case of the revolute joint, the Jacobian matrix of constraints and the right-hand side of the velocity and acceleration constraint equation can be obtained as follows

$$\boldsymbol{\Phi}_{\mathbf{q}}^{(t,2)} = \begin{bmatrix} y_i^P - y_i^Q & -x_i^P + x_i^Q & -(x_j^P - x_i)(x_i^P - x_i^Q) - (y_j^P - y_i)(y_i^P - y_i^Q) \\ 0 & 0 & 1 \\ -y_i^P + y_i^Q & x_i^P - x_i^Q & (x_j^P - x_j)(x_i^P - x_i^Q) + (y_j^P - y_j)(y_i^P - y_i^Q) \\ 0 & 0 & -1 \end{bmatrix} \tag{68}$$

$$\boldsymbol{\upsilon}^{(t,2)} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{69}$$

$$\boldsymbol{\gamma}^{(t,2)} = \begin{Bmatrix} -2[(x_i^P - x_i^Q)(\dot{x}_i - \dot{x}_j) + (y_i^P - y_i^Q)(\dot{y}_i - \dot{y}_j)]\dot{\phi}_i - \\ [(x_i^P - x_i^Q)(y_i - y_j) - (y_i^P - y_i^Q)(x_i - x_j)]\dot{\phi}_i^2 \\ 0 \end{Bmatrix} \tag{70}$$

Constraint Equations

The constraint equations that describe the kinematic restrictions between two bodies can be presented in a simpler manner, that is, they can be substituted by other simple constraint equations when one of the bodies is considered to be fixed, as it is the case of body 1 in the human leg model illustrated in Fig. 36.

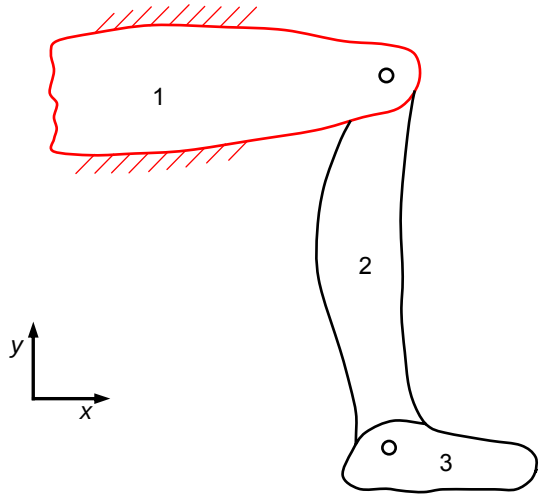


Fig. 36 Human leg multibody model

Thus, in this model, it is clear that body 1 can be constrained by considering three simple constraint equations that restrain their translation and rotation motions, which can be written in the form

$$\mathbf{\Phi}^{(s,1)} \equiv x_1 - c_1 = 0 \tag{71}$$

$$\mathbf{\Phi}^{(s,1)} \equiv y_1 - c_2 = 0 \tag{72}$$

$$\mathbf{\Phi}^{(s,1)} \equiv \phi_1 - c_3 = 0 \tag{73}$$

where c_1, c_2 and c_3 are constant parameters. It is evident that these three constraint equations have null contributions to the right-hand side of the velocity and acceleration constraint equations.

Constraint Equations

Let now consider the slider-crank mechanism represented in Fig. 37, in which the slider (body 4) is constrained to the ground by using a translational joint that can move parallel to the *x*-axis. This kinematic joint can be modeled by considering Eq. (67) associated with a general translational joint.

Since the slider can not move in the *y* direction, and it can not rotate either, then two simple constraint equations can be used stating that the values of variables y_4 and ϕ_4 are constant (Nikravesh 1988).

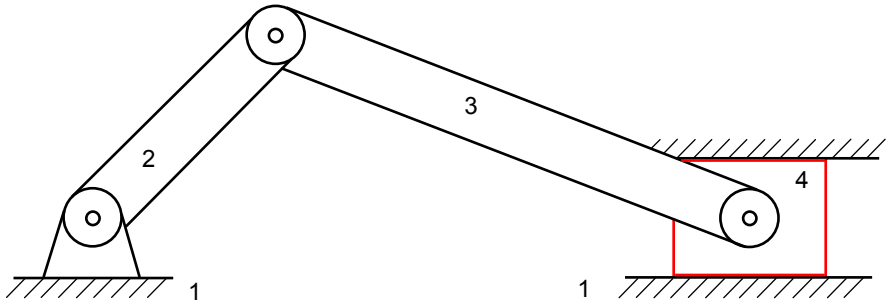


Fig. 37 Slider-crank mechanism

In a broad sense, to constrain the motion of a point P_i on a body *i* in the *x* direction, the following simple constraint equation can be used

$$\mathbf{\Phi}^{(s,1)} \equiv x_i^P - c_4 = 0 \tag{74}$$

or in the expanding form

$$\mathbf{\Phi}^{(s,1)} \equiv x_i + \xi_i^P \cos\phi_i - \eta_i^P \sin\phi_i - c_4 = 0 \tag{75}$$

in which c_4 is a constant parameter.

Constraint Equations

The Jacobian terms of the constraint equation (75) and the right-hand side of velocity and acceleration constraint equations are given by

$$\mathbf{\Phi}_{\mathbf{q}}^{(s,1)} = \left\{1 \quad 0 \quad -\xi_i^P \sin\phi_i + \eta_i^P \cos\phi_i\right\}^T \tag{76}$$

$$\boldsymbol{\upsilon}^{(s,1)} = \{0\} \tag{77}$$

$$\boldsymbol{\gamma}^{(s,1)} = \left\{(-\xi_i^P \cos\phi_i + \eta_i^P \sin\phi_i)\dot{\phi}_i^2\right\} \tag{78}$$

The kinematic constraint equations presented above are functions of the generalized coordinates of the system only, that is, they do not depend on time. In other words, such constraints characterize the physical structure of the multibody systems, being usually provided one or more degrees of freedom that allow the motion of the system. Then, to perform kinematic analysis, the motion of the driving bodies must be specified by using driver elements in the form of driving constraint equations. For instance, in the slider-crank mechanism of Fig. 37, the driving body 2 can rotate according to the following driving constraint equation

$$\mathbf{\Phi}^{(d,1)} \equiv \phi_2 - d_1(t) = 0 \tag{79}$$

where the time function $d_1(t)$ is given by

$$d_1(t) = \frac{1}{2}\alpha_2 t^2 + \omega_2 t + \phi_2^0 \tag{80}$$

in which ω_2 and α_2 are the angular velocity and acceleration of the crank.

Constraint Equations

It can be observed that the contributions of this driving constraint equation to the Jacobian matrix of the constraint and to the right-hand side of the velocity and acceleration constraint equations are expressed as

$$\mathbf{\Phi}_{\mathbf{q}}^{(d,1)} = \begin{Bmatrix} 0 & 0 & 1 \end{Bmatrix}^T \tag{81}$$

$$\boldsymbol{\upsilon}^{(d,1)} = \{-\alpha_2 t - \omega_2\} \tag{82}$$

$$\boldsymbol{\gamma}^{(d,1)} = \{-\alpha_2\} \tag{83}$$

The interested reader in the detailed formulation of other type of driving constraint equations is referred to the works by Nikravesh (1988)

Another type of rehonomic constraint that can be frequently used in the formulation of MBS is the guiding constraint, which imposes the trajectories to certain points on the bodies, as it is the case of the human arm model illustrated in Fig. 38. This type of trajectories is obtained experimentally by using a complete data acquisition system (Silva 2003), being the data points expressed as functions of time variable. Considering, for instance, the trajectory of the center of mass of body 2, then three constraints can be written as

$$\mathbf{\Phi}^{(g,3)} \equiv \begin{Bmatrix} x_2 - t_2^x(t) \\ y_2 - t_2^y(t) \\ \phi_2 - t_2^\phi(t) \end{Bmatrix} = \mathbf{0} \tag{84}$$

where $t_2^k(t)$, (k=x, y, ϕ), represents the trajectory coordinates of the body 2 center of mass (Meireles 2007).

Constraint Equations

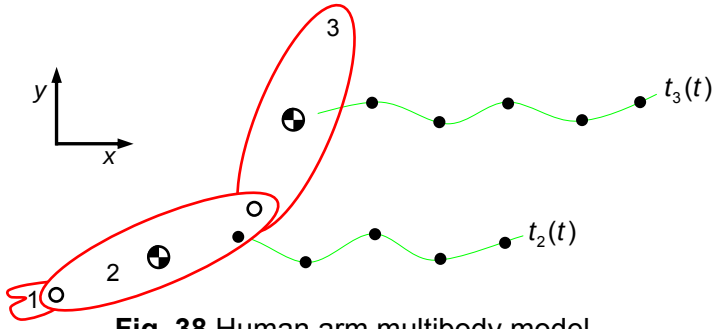


Fig. 38 Human arm multibody model

The contributions of the guiding constraint equations to the Jacobian matrix of the constraints and to the right-hand side of the velocity and acceleration constraints can be evaluated as follows

$$\mathbf{\Phi}_{\mathbf{q}}^{(g,3)} = \begin{bmatrix} \dfrac{\partial \mathbf{\Phi}^{(g,1st)}}{\partial x_2} & \dfrac{\partial \mathbf{\Phi}^{(g,1st)}}{\partial y_2} & \dfrac{\partial \mathbf{\Phi}^{(g,1st)}}{\partial \phi_2} \\ \dfrac{\partial \mathbf{\Phi}^{(g,2nd)}}{\partial x_2} & \dfrac{\partial \mathbf{\Phi}^{(g,2nd)}}{\partial y_2} & \dfrac{\partial \mathbf{\Phi}^{(g,2nd)}}{\partial \phi_2} \\ \dfrac{\partial \mathbf{\Phi}^{(g,3rd)}}{\partial x_2} & \dfrac{\partial \mathbf{\Phi}^{(g,2nd)}}{\partial y_2} & \dfrac{\partial \mathbf{\Phi}^{(g,3rd)}}{\partial \phi_2} \end{bmatrix} \tag{85}$$

$$\boldsymbol{\upsilon}^{(g,3)} = \left\{ \frac{\partial \mathbf{\Phi}^{(g,1st)}}{\partial t} \quad \frac{\partial \mathbf{\Phi}^{(g,2nd)}}{\partial t} \quad \frac{\partial \mathbf{\Phi}^{(g,3rd)}}{\partial t} \right\}^T \tag{86}$$

$$\boldsymbol{\gamma}^{(g,3)} = \left\{ \frac{\partial^2 \mathbf{\Phi}^{(g,1st)}}{\partial t^2} \quad \frac{\partial^2 \mathbf{\Phi}^{(g,2nd)}}{\partial t^2} \quad \frac{\partial^2 \mathbf{\Phi}^{(g,3rd)}}{\partial t^2} \right\}^T \tag{87}$$

Constraint Equations

Since the body's trajectory of center of mass is typically discrete and obtained experimentally, in general, the data collected are used to derive the mathematical expressions by interpolating the coordinates along time.

In the present work, this procedure is obtained by employing cubic splines interpolation, because higher order polynomials tend to swing through wild oscillations in the vicinity of an abrupt change, whereas cubic splines provides much more smooth transitions (Chapra and Canale 1989, Späth 1995).

Furthermore, the use of cubic splines is quite useful to ensure the continuity of the first and the second derivatives, velocity and acceleration respectively, property that is extremely important in the kinematic and dynamic analysis procedures, as Fig. 39 illustrates.

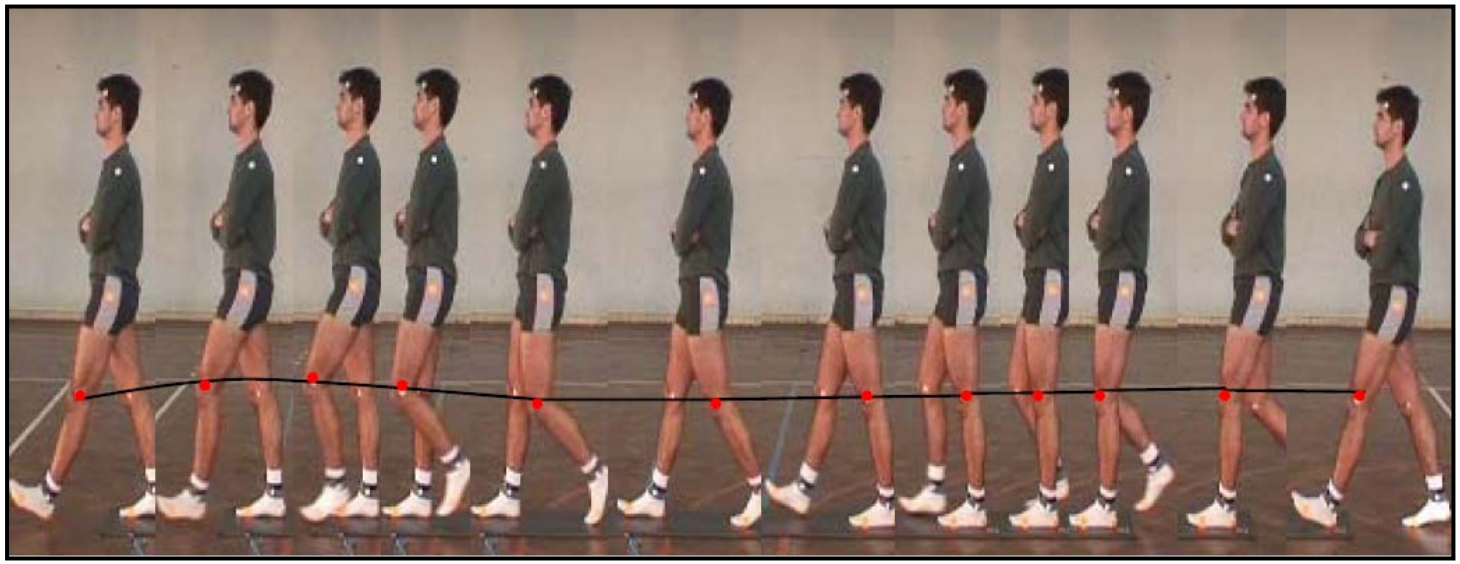


Fig. 39 Example of human gait data acquisition for the knee trajectory

Generalities and System of Linear Equations

As it was stated above the kinematic analysis of a multibody system is performed when the set of equations that result from the kinematic, driving and guiding constraints are assembled and solved to obtain the positions, velocities and accelerations of the system.

On the one hand, the constraint equations that represent the positions of the system are, generally, non-linear in terms of **q**, reason why iterative schemes are required to solve them, such as the popular and efficient Newton-Raphson method.

On the other hand, the velocity and acceleration constraint equations represent linear systems in terms of $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$, respectively, that can be solved by employing any usual method adopted to deal with systems of linear equations, such as the Gaussian elimination method.

The main purpose of this section is to present the elementary numerical methods used in the kinematic analysis of multibody systems when appropriate driving and guiding conditions are specified.

The reader who is interested in more details is referred to the specialized thematic literature (Nikravesh 1988, Haug 1989, Amirouche 1992, Jálon and Bayo 1994, Eich-Soellner and Führer 1998).

A system of *n* linear algebraic equations in n unknowns with real coefficients can be expressed in the following form (Conte and Boor 1981)

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \qquad\qquad \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases} \tag{88}$$

which has one and only one solution for every right-hand side.

System of Linear Equations

The system of equations (88) can be re-written in a compact form as

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{89}$$

where **A** is the matrix of coefficients, **x** represents the vector of the unknowns and **b** is the vector that contains the independent terms

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ & & \dots & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \tag{90}$$

$$\mathbf{x} = \begin{Bmatrix} x_1 & x_2 & \dots & x_n \end{Bmatrix}^T \tag{91}$$

$$\mathbf{b} = \begin{Bmatrix} b_1 & b_2 & \dots & b_n \end{Bmatrix}^T \tag{92}$$

The solution of the system of equations (88) is a vector **x** that verifies simultaneously the system of *n* linear equations.

System of Linear Equations

A system of linear equations is said to be determined when it has a unique solution. In other words, the matrix **A** has to be invertible, which, in turn, requires that the determinant of matrix **A** be different from zero.

When the determinant of matrix **A** is not zero, then **A** is said to be nonsingular and there exists the inverse matrix $\mathbf{A}^{-1}$, such that $\mathbf{A}^{-1}\mathbf{A}=\mathbf{A}\mathbf{A}^{-1}=\mathbf{I}$.

The numerical methods commonly used to solve linear systems can be classified into two groups, namely the direct and iterative methods (Dahlquist and Björck 1974).

The first group includes the methods that, in absence of round-off or other errors, yield the exact solution after a finite number of elementary operations, such as multiplication of one equation by a nonzero constant, addition of a multiple of one equation to another equation and interchange of two equations.

Iterative methods give a sequence of approximate solutions, converging when the number of steps tends to infinity.

The choice between direct and iterative methods depends on the proportion and distribution as well as sign and size of the nonzero elements of **A**.

Matrices associated with linear systems can also be classified as dense or sparse.

Dense matrices have very few zero elements. It is usually most efficient to handle problems involving dense matrices by direct methods.

In turn, sparse matrices have very few nonzero elements, being, in general, more appropriate to use iterative methods with this type of matrices.

The most important and efficient among the direct methods to solve linear systems are the Gaussian elimination and the LU-factorization methods.

These two methods are presented and discussed in the following paragraphs.

System of Linear Equations

In a broad sense, the idea behind the Gaussian elimination method is to eliminate the unknowns in a systematic manner, in order to obtain a triangular system, solution of which is trivial.

A first phase of the Gaussian elimination method consists of a forward elimination, in which the matrix **A** is converted into an upper-triangular matrix.

Then, by using a back substitution procedure the system is solved for the unknowns **x**. One possible approach to use the Gaussian elimination method consists of converting the matrix **A** into an upper-triangular matrix that has '1s' on the diagonal.

In order to better understand this procedure let consider the system of linear equations given by (93), which purpose is to find the values of the unknowns x_1, x_2 and x_3 (Nikravesh 1988).

$$\begin{bmatrix} 3 & 1 & -1 \\ -1 & 2 & 1 \\ 2 & -3 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 2 \\ 6 \\ -1 \end{Bmatrix} \tag{93}$$

The first step consists of dividing the first equation by 3 yielding

$$\begin{bmatrix} 1 & 1/3 & -1/3 \\ -1 & 2 & 1 \\ 2 & -3 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 2/3 \\ 6 \\ -1 \end{Bmatrix} \tag{94}$$

Adding the first and second equation, and then add -2 times the first equation to the third results in

System of Linear Equations

$$\begin{bmatrix} 1 & 1/3 & -1/3 \\ 0 & 7/3 & 2/3 \\ 0 & -11/3 & 5/3 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 2/3 \\ 20/3 \\ -7/3 \end{Bmatrix} \tag{95}$$

Multiplying now the second equation by 3/7 yields

$$\begin{bmatrix} 1 & 1/3 & -1/3 \\ 0 & 1 & 2/7 \\ 0 & -11/3 & 5/3 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 2/3 \\ 20/7 \\ -7/3 \end{Bmatrix} \tag{96}$$

Adding 11/3 times the second equation to the third equation results in

$$\begin{bmatrix} 1 & 1/3 & -1/3 \\ 0 & 1 & 2/7 \\ 0 & 0 & 19/7 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 2/3 \\ 20/7 \\ 57/7 \end{Bmatrix} \tag{97}$$

Finally, multiplying the third equation by 7/19 yields

$$\begin{bmatrix} 1 & 1/3 & -1/3 \\ 0 & 1 & 2/7 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 2/3 \\ 20/7 \\ 3 \end{Bmatrix} \tag{98}$$

System of Linear Equations

At this stage it must be noted that all of the elements below the diagonal of the matrix are zero, consequently, the determinant of the matrix is 1, and, therefore, it is nonsingular.

Applying now a back substitution, the solution of the linear system (93) can easily be obtained. Thus, from the third equation yields

$$x_3 = 3 \tag{99}$$

from the second equation results that

$$x_2 + \frac{2}{7}(3) = \frac{20}{7} \Rightarrow x_2 = 2 \tag{100}$$

and, finally, using the first equation yields

$$x_1 + \frac{1}{3}(2) - \frac{1}{3}(3) = \frac{2}{3} \Rightarrow x_1 = 1 \tag{101}$$

In many multibody applications it is usual to deal with the problem to solve several systems of linear equations that have a common matrix of coefficients, being different the vectors of independent terms. Equations (44) and (45) are examples of these circumstances in the measure that they represent linear systems that have a common Jacobian matrix. In these cases, the LU-factorization method is quite attractive in terms of efficiency and computer memory utilization. This method consists of decomposition of any nonsingular matrix **A** into a lower-triangular **L** and an upper-triangular matrix **U** such that

$$\mathbf{A} = \mathbf{LU} \tag{102}$$

System of Linear Equations

It is clearly evident why the factorization of matrix **A** into the product **LU** is called LU-factorization. As a consequence of this procedure, Eq. (89) is equivalent to

$$\mathbf{Ax} = \mathbf{LUx} = \mathbf{b} \tag{103}$$

which can be decomposed into two triangular systems as

$$\mathbf{Ly} = \mathbf{b} \tag{104}$$

$$\mathbf{Ux} = \mathbf{y} \tag{105}$$

Thus, if **L** and **U** are known, then Eq. (104) can be solved for **y** and then Eq. (105) can be solved for **x**, by back substitution because both have triangular coefficient matrices. It should be noted that the resolution of triangular systems involves a small number of arithmetic operations. Furthermore, the LU-factorization supposes that matrix **A** can be decomposed in a unique manner according to Eq. (102).

The Crout's method is one of the most popular approaches to calculate the elements **L** and **U**, which solves this problem in a recursively manner. This method can be better illustrated through an example. For this purpose, let consider a matrix **A** of rank *n*=3, that does not require row or column interchanges during the process, written as follows

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{106}$$

System of Linear Equations

An auxiliary matrix **B** that contains elements **L** and **U**, can be defined as

$$\mathbf{B} = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ l_{21} & u_{22} & u_{23} \\ l_{31} & l_{32} & u_{33} \end{bmatrix} \tag{107}$$

The elements of **L** and **U** can be calculated by equating Eqs. (106) and (107) yielding the following relations

$$u_{11} = a_{11} \tag{108}$$

$$u_{12} = a_{12} \tag{109}$$

$$u_{13} = a_{13} \tag{110}$$

$$l_{21} = a_{21}/u_{11} \tag{111}$$

$$l_{31} = a_{31}/u_{11} \tag{112}$$

$$u_{22} = a_{22} - l_{21}/u_{12} \tag{113}$$

$$u_{23} = a_{23} - l_{21}/u_{13} \tag{114}$$

$$l_{32} = (a_{32} - l_{31}\mathrm{u}_{12})/u_{22} \tag{115}$$

$$u_{33} = a_{33} - l_{31}\mathrm{u}_{13} - l_{32}u_{23} \tag{116}$$

System of Linear Equations

Applying the LU-factorization process to the following matrix **A**

$$\mathbf{A} = \begin{bmatrix} 2 & 2 & 1 \\ 1 & -3 & 2 \\ -1 & 1 & -1 \end{bmatrix} \tag{117}$$

yields

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ -1/2 & -1/2 & 1 \end{bmatrix} \tag{118}$$

$$\mathbf{U} = \begin{bmatrix} 2 & 2 & 1 \\ 0 & -4 & 3/2 \\ 0 & 0 & 1/4 \end{bmatrix} \tag{119}$$

The Gaussian elimination and LU-factorization methods are equivalents and, in general, these methods are carried out with refinements such as the necessity for pivoting schemes as well as the treatment of nonsquare matrices.

These issues are out of the scope of the present work, being the interested reader referred to the thematic literature, where different approaches are analyzed and discussed, being also offered some specific routines (Dahlquist and Björck 1974, Conte and Boor 1981).

Nonlinear System of Equations

The kinematic position analysis of multibody systems deals with the solution of a set of nonlinear algebraic equations, that is, it consists of finding the roots of the position constraint equations

$$\mathbf{\Phi}(\mathbf{q},t)=\mathbf{0} \tag{120}$$

in which, in general, it is not possible to obtain the explicit form of the constraint equations to find their roots. Therefore, it is required to use numerical iterative procedures, such as the well known Newton-Raphson method to obtain approximate solutions.

Due to its good computational efficiency, this method is one of the most popular and frequently method used to solve the set of nonlinear algebraic equations that represent the position kinematic constraints (Nikravesh 1988, Haug 1989). The implementation of the Newton-Raphson method requires the knowledge of an initial guess or estimate for the desired solution, being then generated a sequence of appropriate solutions that, under certain conditions, converge to the desired root. Additionally, it is also necessary to specify a tolerance that defines the stop criterion in this iterative process (Dahlquist and Björck 1974, Pina 1995).

With the purpose to better understand the main issues associated with the iterative Newton-Raphson method, let consider the case of a one-dimensional nonlinear rehonomic constraint equation expressed as

$$\Phi(q)=0 \tag{121}$$

where the solution of q is to be determined. The Newton-Raphson method can be expressed as

$$q^{i+1}=q^{i}-\frac{\Phi(q^{i})}{\Phi_{q}(q^{i})} \tag{122}$$

in which $\Phi_q(q^i)$ represents the Jacobian matrix evaluated at $q=q^i$.

Nonlinear System of Equations

At this stage, it should be highlighted that the Newton-Raphson method has the drawback of demanding for the analytical evaluation of the first derivative of function $\Phi(q)$.

Equation (122) results from the fact of approximate the given function to a straight-line tangent to the function at a given point q^i and evaluate the intersection of that straight-line with the q-axis. This procedure results in the root of the straight-line.

Figure 40 illustrates the graphical interpretation of the Newton-Raphson method for the two first iterations and for an example in which the method converges to the desired solution after a finite number of iterations.

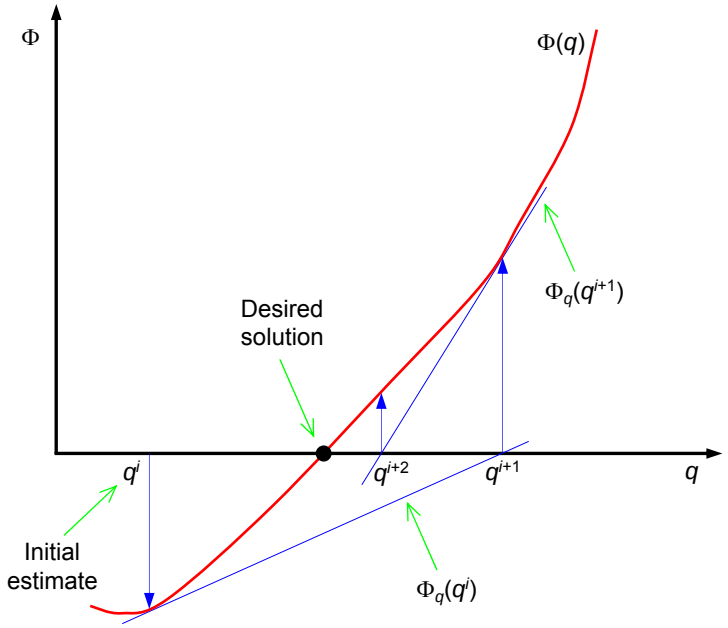


Fig. 40 Graphical representation of the Newton-Raphson method

Nonlinear System of Equations

Equation (122) can be re-written as

$$\Phi_q(q^i)(q^{i+1} - q^i) = -\Phi(q^i) \tag{123}$$

which can alternatively be expressed as

$$\Phi_q(q^i)\Delta q^i = -\Phi(q^i) \tag{124}$$

where

$$q^{i+1} = q^i + \Delta q^i \tag{125}$$

The computational strategy for the iterative Newton-Raphson method shown in Fig. 41 can be described as:

1. Specify the maximum number of allowed iterations, Nr_{max}, define a tolerance, *tol*, and initialize the iteration counter *i*, *i*=0.
2. Define the initial estimate q^i.
3. Evaluate functions $\Phi(q^i)$ and $\Phi q(q^i)$.
4. Solve Eq. (124) for Δq^i.
5. Calculate $qi+1$ by solving Eq. (125).
6. If $|q^{i+1}–q^i|<tol$, then the method ends, otherwise, go to step 4) by incrementing *i*, i.e., *i*=*i*+1.

Nonlinear System of Equations

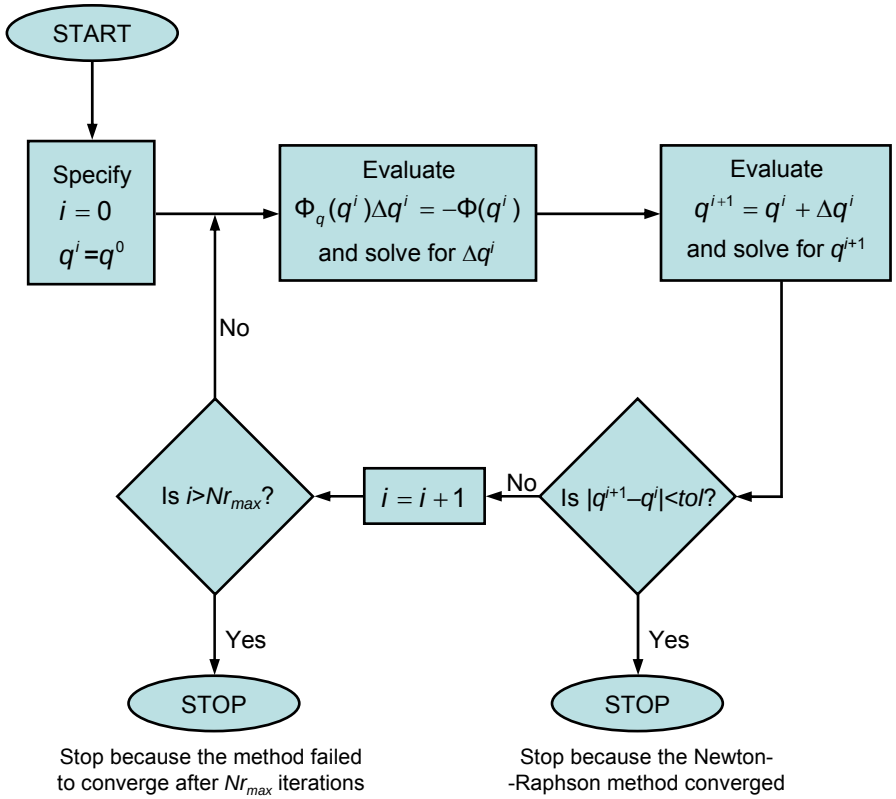


Fig. 41 Flowchart of computational procedure for the Newton-Raphson method

Thus, it is clear that Eq. (124) represents a linear system that is solved for Δqi and, subsequently, $qi+1$ is evaluated directly by solving Eq. (125). The term Δqi is known as Newton difference and represents the difference of the approximate solution at the iteration *i*.

The solution of the linear system of Eq. (124) can be obtained by using the Gaussian elimination method or LU-factorization scheme presented previously.

Nonlinear System of Equations

In spite of its very good computational efficiency, the Newton-Raphson method has some limitations, being some of them represented in Fig. 42.

In Fig. 42a, the Newton-Raphson method fails because the desired solution is located at an inflection point.

The problem of multiple solutions of a function is illustrated in Fig. 42b. In this case, the initial estimate influences the root obtained.

Figure 42c shows the divergence when the initial estimate is near to a local minimum.

Finally, Fig. 42d represents the situation where the tangent to the function corresponding at the initial estimate is parallel to the q-axis, which means that the function $\Phi_q(q^i)$ assumes a zero value.

Thus, it is crucial to select a reasonable estimate of the solution to be used to initiate the iterative process (Nikravesh 1988, 2007).

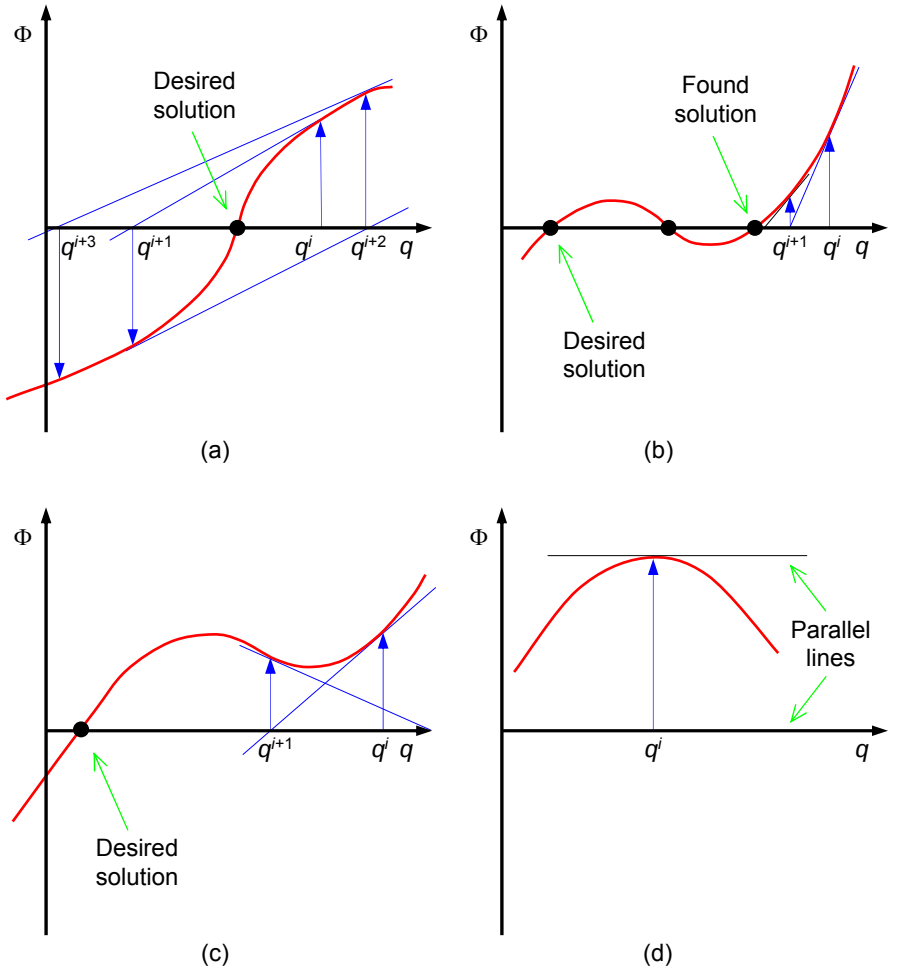


Fig. 42 Examples in which the Newton-Raphson method fails to converge

Nonlinear System of Equations

In multibody systems, the fact of having multiple solutions means that it is possible to have different configurations for the same input variable, as it is the case illustrated in Fig. 43.

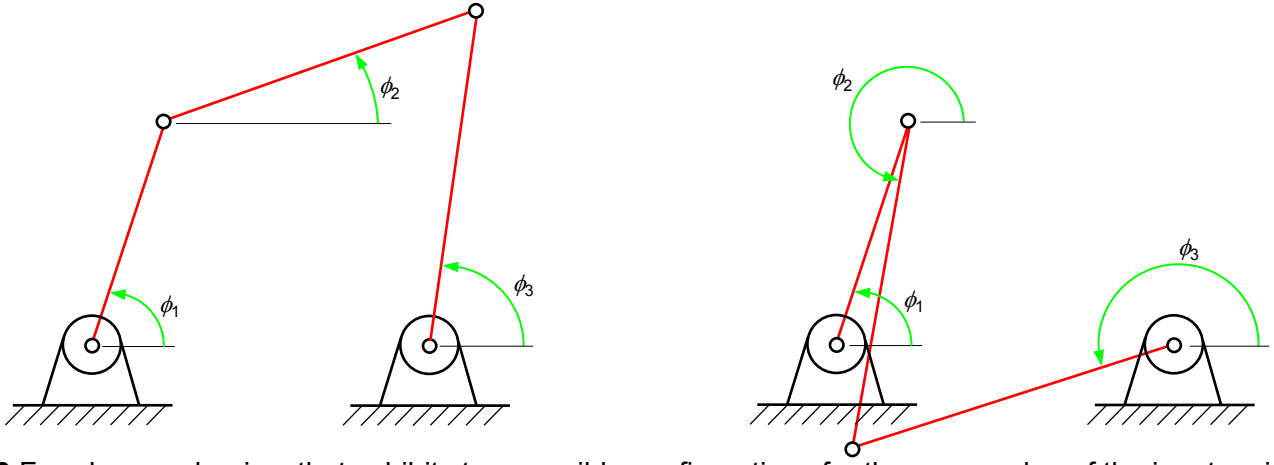


Fig. 43 Four-bar mechanism that exhibits two possible configurations for the same value of the input variable ϕ_1

The Newton-Raphson does not converge when the function $\Phi_q(q^i)$ does not have solution, as in the case of the four bar mechanism of Fig. 44.

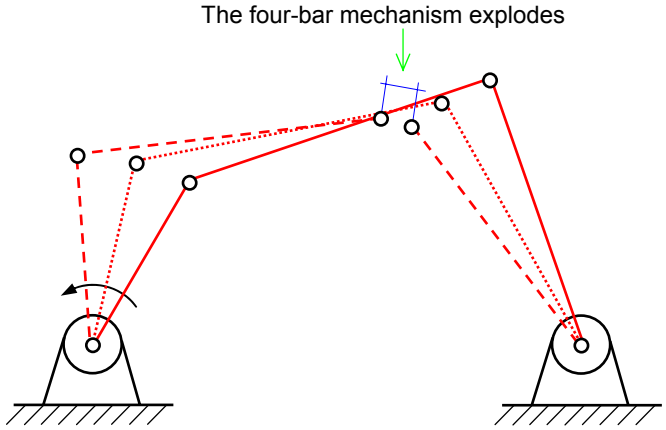


Fig. 44 Double-rocker four-bar mechanism where the Newton-Raphson method fails to converge since the function $\Phi_q(q^i)$ have no solution

Nonlinear System of Equations

Finally, Fig. 45 shows a slider-crank mechanism in which the iterative Newton-Raphson method does not converge. In this particular multibody system, the lengths of the crank and connecting rod are equal to each other. Thus, when the crank and connecting rod become coincident, the system response can be as in the case represented by dashed lines or by points.

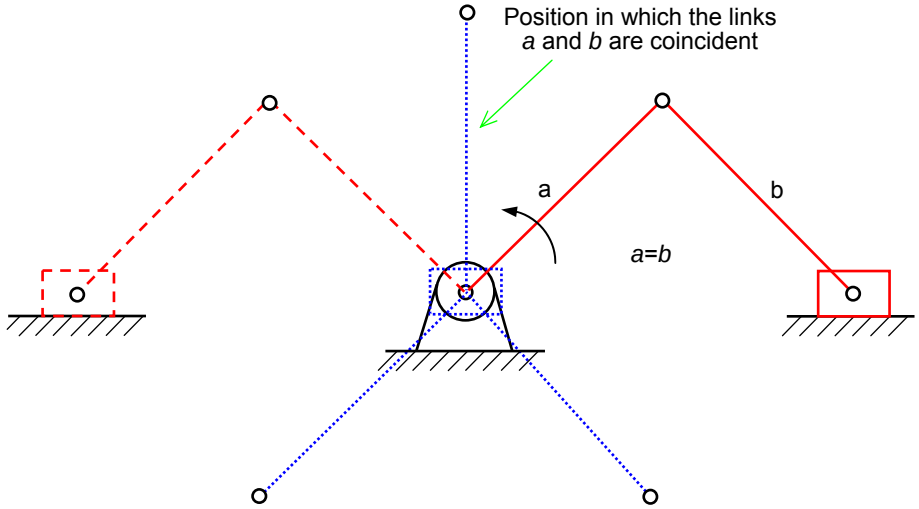


Fig. 45 Slider-crank mechanism in which the Newton-Raphson method fails

Forces present in MBS

There are many types of forces that can be present in multibody systems, such as (Flores 2008)

- gravitational forces,
- inertia forces,
- spring-damper-actuator forces,
- normal contact forces,
- tangential or frictional forces,
- external applied forces,
- forces due to elasticity of bodies,
- and thermal, electrical and magnetic forces.

However, only the first seven types of forces are relevant in the multibody systems of common application. Figure 46 shows a mechanical system composed by six links and a free sphere that can collide with the slider (body 6). In this system, it is possible to identify the main types of forces described above.

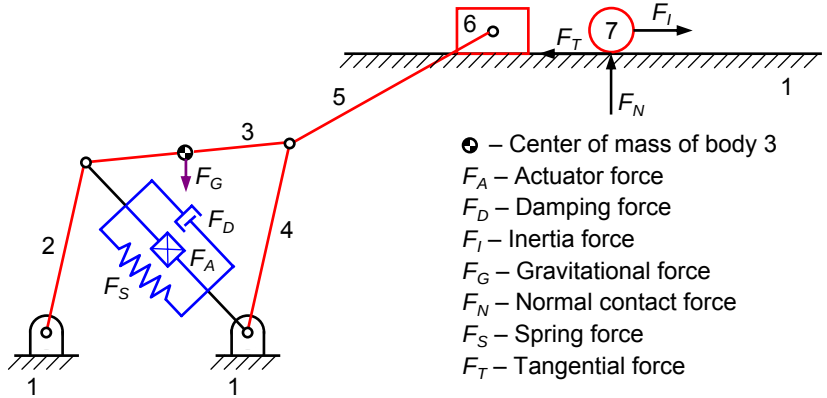


Fig. 46 Different types of forces that can be present in a mechanical system

Forces present in MBS

In a broad sense, the forces in the multibody systems can be divided into two main groups, namely external and internal forces. If the forces are located inside of the boundaries of the system, they are called internal, otherwise the forces are named external forces. Figure 47 illustrates a MBS in which the internal and external forces can easily be identified. The definition of the system boundaries depend on the system in analysis.

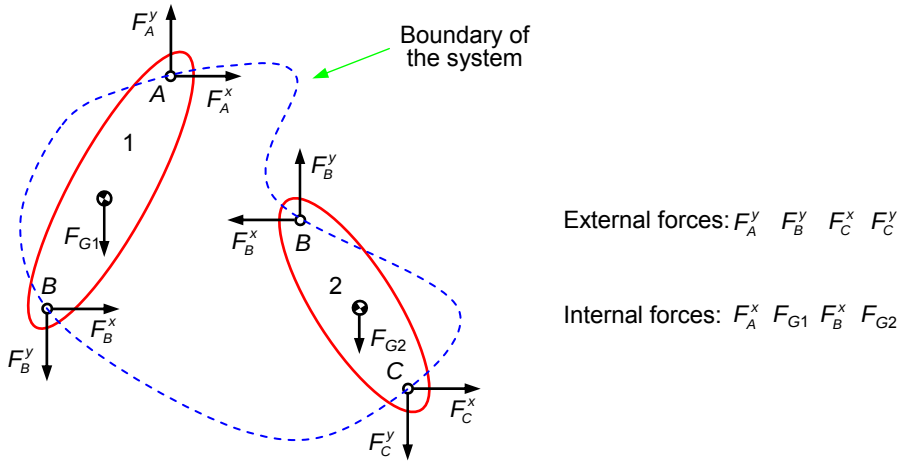


Fig. 47 External and internal forces in a mechanical system

Figure 48 illustrates a body *i* acted upon by a gravitational field in the negative *y* direction. The choice of the negative *y* direction as the direction of gravity is totally arbitrary. However, in the present work, the gravitational field will be considered to be acting in this direction unless indicated otherwise. The force due to gravitational field can be written as

$$\mathbf{F}_G = -m_i g \vec{\mathbf{u}}_y \tag{126}$$

where m_i is the mass of body *i* and *g* is gravity acceleration. In the present work the acceleration due to gravity assumes a value equal to 9.81 m/s^2.

Forces present in MBS

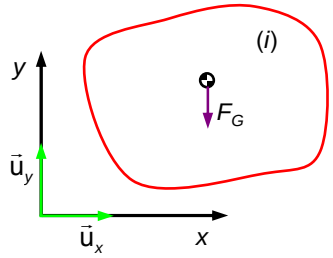


Fig. 48 Gravitational field acting on a body *i*

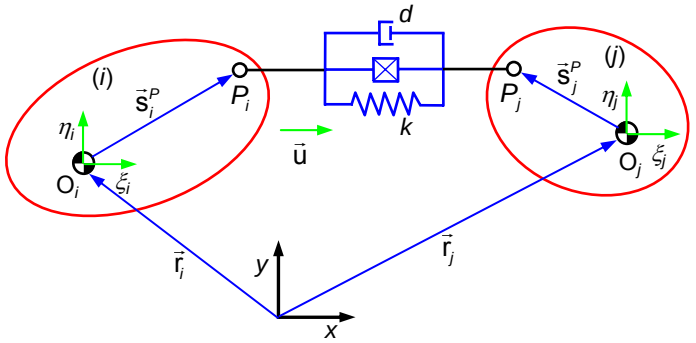


Fig. 49 Spring-damper-actuator element connecting bodies *i* and *j*

The module of the gravitational force is given by

$$F_G = m_i g \tag{127}$$

Figure 49 shows a spring-damper-actuator element connecting bodies *i* and *j* through two points of connectivity P_i and P_j, whose local position vectors with respect to the origins of their respective local coordinate systems are $\mathbf{s}'^P_i$ and $\mathbf{s}'^P_j$. The spring-damper-actuator force is given by

$$\mathbf{F}_{SDA} = \left[k(l - l^0) + d\dot{l} + F_A \right] \mathbf{u} \tag{128}$$

where the first term on the right-hand side is the spring force, the second term represents the damper force and the third term denotes the actuator force. The spring stiffness is represented by *k*, *l* is the deformed length, l^0 is the undeformed or natural of the spring, *d* is the damping coefficient of the damper and $\dot{l}$ is the time rate of change of the damper length.

Forces present in MBS

The vector *l* that connects the points P_i and P_j can be evaluated by

$$\mathbf{l} = \mathbf{r}_i + \mathbf{A}_i \mathbf{s}'^P_i - \mathbf{r}_j - \mathbf{A}_j \mathbf{s}'^P_j \tag{129}$$

The magnitude of this vector is

$$l = \sqrt{\mathbf{l}^T \mathbf{l}} \tag{130}$$

The unit vector along the spring-damper-actuator element is defined as

$$\mathbf{u} = \frac{\mathbf{l}}{l} \tag{131}$$

The time rate of change of the damper length can be obtained by differentiating Eq. (130), yielding

$$\dot{l} = \frac{\mathbf{l}^T \dot{\mathbf{l}}}{l} \tag{132}$$

where $\dot{\mathbf{l}}$, in turn, is found from Eq. (129)

$$\dot{\mathbf{l}} = \dot{\mathbf{r}}_i + \dot{\phi}_i \mathbf{B}_i \mathbf{s}'^P_i - \dot{\mathbf{r}}_j - \dot{\phi}_j \mathbf{B}_j \mathbf{s}'^P_j \tag{133}$$

in which

$$\mathbf{B}_k = \begin{bmatrix} -\sin\phi_k & -\cos\phi_k \\ -\cos\phi_k & \sin\phi_k \end{bmatrix} \qquad (k=i, j) \tag{134}$$

Forces present in MBS

Figure 50 shows two spheres that contact with each other and where a relative penetration δ is verified, which can be evaluated from the system kinematics. The normal and tangential contact forces can be expressed as

$$F_N = K\delta^n \tag{135}$$

$$F_T = -\mu F_N \tag{136}$$

where Eq. (135) refers to the Hertz contact force and Eq. (136) is the well known Coulomb's friction law (Flores et al. 2006). The variable K is a generalized stiffness parameter and μ represents the coefficient of friction. The problem of contact force models in the context of multibody dynamics is out of scope of this work.

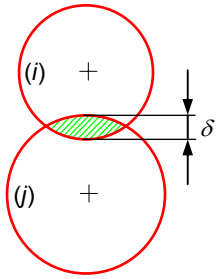


Fig. 50 Penetration between two contacting spheres *i* and *j*

Equations of Motion

The formulation of the equations of motion for constrained multibody system adopted here follows closely the Nikravesh's work, in which the generalized absolute coordinates are used to describe the system configuration (Nikravesh 1988).

For a constrained multibody system the kinematic constraints can be described by a set of linear and/or nonlinear holonomic algebraic equations as

$$\mathbf{\Phi}(\mathbf{q}, t) = \mathbf{0} \tag{137}$$

where **q** is the vector of generalized coordinates and *t* is the time variable. The set of equations of motion for a constrained multibody system can be represented by (Nikravesh 1988)

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{q}}^{T}\boldsymbol{\lambda} = \mathbf{g} \tag{138}$$

in which **M** is the system mass matrix, $\ddot{\mathbf{q}}$ is the vector that contains the system accelerations, $\mathbf{\Phi}_{\mathbf{q}}$ is the Jacobian matrix of the constraint equation (137), $\boldsymbol{\lambda}$ is the Lagrange multipliers vector and **g** is the generalized forces vector.

In order to progress with solution, the constraint velocity and acceleration equations are required. Thus, differentiating Eq. (137) with respect to time yields the velocity constraint equations

$$\mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} = -\mathbf{\Phi}_{t} = \boldsymbol{\upsilon} \tag{139}$$

in which $\dot{\mathbf{q}}$ is the vector of generalized velocities and $\boldsymbol{\upsilon}$ is the right-hand side of velocity equations.

Equations of Motion

A second differentiation of Eq. (137) with respect to time leads to the acceleration constraint equations

$$\boldsymbol{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}} = -(\boldsymbol{\Phi}_{\mathbf{q}}\dot{\mathbf{q}})_{\mathbf{q}}\dot{\mathbf{q}} - 2\boldsymbol{\Phi}_{\mathbf{q}t}\dot{\mathbf{q}} - \boldsymbol{\Phi}_{tt} = \boldsymbol{\gamma} \tag{140}$$

in which $\ddot{\mathbf{q}}$ is the acceleration vector and $\boldsymbol{\gamma}$ is the right-hand side of acceleration equations. It is known that in the dynamic analysis, Eqs. (137), (139) and (140) must be satisfied during the analysis.

Thus, Eq. (140) can be appended to Eq. (138) and re-written in matrix form as

$$\begin{bmatrix} \mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^{T} \\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{g} \\ \boldsymbol{\gamma} \end{Bmatrix} \tag{141}$$

Equation (141) is formed as a combination of the equations of motion and kinemtaic constraint equations, often referred to as a mixed set of differential and algebraic equations (DAE's).

This system of equations is solved for $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$. Then, in each integration time step, the accelerations vector, $\ddot{\mathbf{q}}$, together with velocities vector, $\dot{\mathbf{q}}$, are integrated in order to obtain the system velocities and positions for the next time step. This procedure is repeated up to final time of analysis is reached.

A set of initial conditions, positions and velocities, is required to start the dynamic simulation. In the present work, the initial conditions are based on the results of kinematic simulation of the mechanical systems. The subsequent initial conditions for each time step in the simulation are obtained in the usual manner from the final conditions of the previous time step (Nikravesh 2007).

Equations of Motion

The system of the motion equations (141) does not use explicitly the position and velocity equations associated with the kinematic constraints, that is, Eqs. (137) and (139).

Consequently, for moderate or long simulations, the original constraint equations start to be violated due to the integration process and/or to inaccurate initial conditions.

Therefore, special procedures must be followed to avoid or minimize this phenomenon.

Several methods to solve this problem have been suggested and tested, being the most common among them the Baumgarte stabilization method (Baumgarte 1972), the augmented Lagrangian formulation (Bayo et al. 1988) and the coordinate partitioning method (Wehage and Haug 1982).

Due to its simplicity and easiness for computational implementation, the Baumgarte stabilization method is the most popular and attractive technique to overcome the drawbacks of the standard integration of the equations of motion.

Baumgarte's method can be looked upon as an extension of feedback control theory.

The principle of this method is to damp out the acceleration constraint violations by feeding back the violations of the position and velocity constraints.

The choice of the feedback parameters depends on several factors, namely, the integrator used and the model of the multibody system.

This method does not solve all possible numerical instabilities as, for instance, those that arise near kinematic singularities.

Furthermore, the major drawback of the Baumgarte's method is the ambiguity in choosing feedback parameters. As pointed out by Baumgarte (1972), it seems that the choice of these coefficients usually involves a trial and error procedure. This method is discussed in detail in section 8.

Equations of Motion

The augmented Lagrangian formulation is based on Hamilton's principle and the constraint equations are taken into account using a penalty approach.

This method consists of solving the system's equations of motion using an iterative process.

The form of the constraint equations is similar to the form proposed in the Baumgarte's method but it has the advantage of handling redundant constraints in the process (Neto and Ambrósio 2003, Flores et al. 2008).

In the coordinate partitioning method, the generalized coordinates are partitioned into independent and dependent sets.

The numerical integration is carried out for independent generalized coordinates.

Then, the constraint equations are solved for dependent generalized coordinates using, for instance, the Newton-Raphson method.

The advantage of this method is that it satisfies all the constraints to the level of precision specified and maintains good error control.

However, it suffers from poor numerical efficiency due to the requirement for the iterative solution for dependent generalized coordinates in the Newton-Raphson method.

During integration, numerical problems may arise due to inadequate choice of independent and dependent coordinates that lead to poorly conditioned matrices.

For details on this methodology, the interested reader is referred to the work by Wehage and Haug (1982).

In fact, this method was originally developed by Maggi in 1903 and later revised by Kane in the sixties (Flores and Seabra 2009).

Equations of Motion

In addition to these three basic approaches, many research papers have been published on the stabilization methods for the numerical integration the equations of motion of multibody systems.

Yoon et al. (1994) presented a direct correction method to eliminate the violation of the constraints in numerical simulation of constrained multibody systems. Yet, this method is formulated on the positions level only.

Blajer (1995) considered the projection method to obtain the dynamic equations of motion for constrained multibody systems in the form of ordinary differential equations.

Then, a standard solver is used to integrate the resulting system. Fisette and Vaneghem (1996), based on the coordinate partitioning method, used the LU-factorization of constraint Jacobian matrix to identify the dependent and independent coordinates.

This aspect is of paramount importance since during the integration process, numerical problems may arise due to inadequate choice of the independent coordinates that lead to poorly conditioned matrices.

This problem was also considered by Arabyan and Wu (1998) to study multibody mechanical systems with both holonomic and nonholonomic constraints.

Weijia et al. (2000) used the Taylor's expansion series to present a methodology to deal with the violation of the constraints.

Neto and Ambrósio (2003) used different methodologies to handle the constraint violation correction for the integration of differential algebraic equations in the presence of redundant constraints.

Tseng et al. (2003) used the Maggi's equations with perturbed iteration to develop an efficient approach to numerically solve constrained multibody systems.

Nikravesh (1984) comparatively studied the direct integration of the equations of motion of multibody systems, the Baumgarte stabilization method and the coordinate partitioning method, and concluded that the implementation of the Baumgarte approach is twice as efficient as the integration of the mixed system.

Equations of Motion

In what follows, the main numerical aspects related to the standard integration of the equations of motion of a multibody system are revisited.

The standard integration of the equations of motion, here called direct integration method (DIM), converts the n second-order differential equations of motion into $2n$ first-order differential equations.

Then, a numerical scheme, such as the fourth-order Runge-Kutta method, is used to solve the initial value problem (Shampine and Gordon 1975, Gear 1981).

The $2n$ differential equations of motion are solved without considering the integration numerical errors and, consequently, during the simulation the propagation of these types of errors results in constraint violations.

The two error sources that lead to constraint violations for any numerical integration step are truncation and round off errors.

Truncation or discretization errors are caused by the nature of the techniques employed to approximate values of a function, **y**.

Round off errors are due to the limited numbers of significant digits that can be retained by a computer.

The truncation errors are composed of two parts:

- the first is a local truncation error that results from the application of the method over a single step,
- and the second is the propagated error that results from the approximation procedure applied in the previous step.

The sum of the two is the total or global truncation errors.

Equations of Motion

The commonly used numerical integration algorithms are useful in solving first-order differential equations that take the form (Gear 1981)

$$\dot{\mathbf{y}} = f(\mathbf{y}, t) \tag{142}$$

Thus, if there are *n* second order differential equations, they are converted to 2*n* first order equations by defining the **y** and $\dot{\mathbf{y}}$ vectors, which contains, respectively, the system positions and velocities and the system velocities and accelerations, as follows

$$\mathbf{y} = \begin{Bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{Bmatrix} \quad \text{and} \quad \dot{\mathbf{y}} = \begin{Bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{Bmatrix} \tag{143}$$

The reason for introducing these new vectors **y** and $\dot{\mathbf{y}}$ is that most numerical integration algorithms deal with first order differential equations (Shampine and Gordon 1975). The following diagram can interpret the process of numerical integration at instant of time *t*

$$\dot{\mathbf{y}}(t) \xrightarrow{\text{Integration}} \mathbf{y}(t + \Delta t) \tag{144}$$

In other words, velocities and accelerations at instant *t*, after integration process, yield positions and velocities at next time step, $t=t+\Delta t$.

Equations of Motion

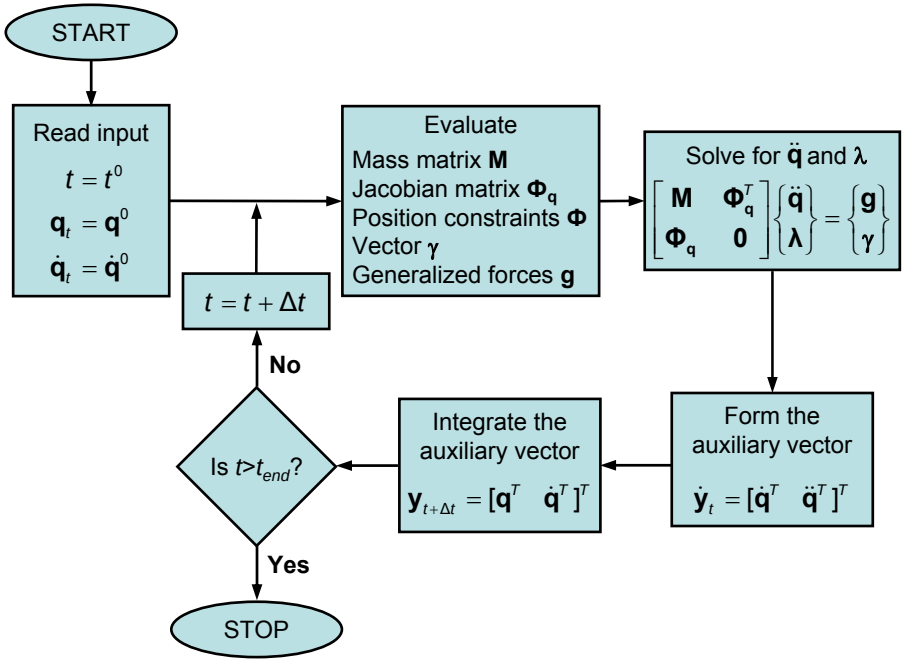


Fig. 51 Flowchart of computational procedure for dynamic analysis of multibody systems based on the direct integration method

Figure 51 presents a flowchart that shows the algorithm of direct integration method of the equations of motion. At $t=t^0$, the initial conditions on $\mathbf{q}^0$ and $\dot{\mathbf{q}}^0$ are required to start the integration process.

These values can not be specified arbitrarily, but must satisfy the constraint equations defined by Eqs. (137) and (139). The direct integration algorithm presented in Fig. 51 can be summarized by the following steps:

1. Start at instant of time t^0 with given initial conditions for positions $\mathbf{q}^0$ and velocities $\dot{\mathbf{q}}^0$.
2. Assemble the global mass matrix **M**, evaluate the Jacobian matrix $\mathbf{\Phi}_\mathbf{q}$, construct the constraint equations $\mathbf{\Phi}$, determine the right-hand side of the accelerations $\boldsymbol{\gamma}$, and calculate the force vector **g**.

Equations of Motion

3. Solve the linear set of the equations of motion (141) for a constrained multibody system in order to obtain the accelerations $\ddot{\mathbf{q}}$ at instant *t* and the Lagrange multipliers $\boldsymbol{\lambda}$.
4. Assemble the vector $\dot{\mathbf{y}}_t$ containing the generalized velocities $\dot{\mathbf{q}}$ and accelerations $\ddot{\mathbf{q}}$ for instant of time *t*.
5. Integrate the $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ vectors for time step $t+\Delta t$ and obtain the new positions and velocities.
6. Update the time variable, go to step 2) and proceed with the process for a new time step, until the final time of analysis is reached.

The direct integration method of equations of motion is prone to integration errors, because the constraint equations (137) and (139) are only satisfied at the initial instant of time.

In the first few time steps, the constraint violations are usually small and negligible.

However, as time progresses, the error in computed values for kinematic parameters is accumulated and constraint violations increase.

Hence, the produced results can be unacceptable; and therefore, the direct integration method requires the use of a constraint stabilization technique, especially for long simulations.

It should be noted that the direct integration method is quite sensitive to initial conditions, which can be an important source of errors in the integration process (Flores et al. 2008).

Generalities

In the previous section, the equations of motion for multibody systems were derived from the Newton-Euler formulation together with the augmentation method.

The Newton-Euler equations represent the translational and rotational motions of bodies, while the augmentation method is used to adjoin the constraint equations of the multibody systems.

In other words, the augmentation formulation denotes the process where the algebraic kinematic constraint equations are augmented to the differential equations of motion, in order that the number of unknowns for which the system is being solved corresponds to the number of system equations (Nikravesh 1988, Haug 1989).

As a consequence, the equations of motion of multi-body systems (141) are differential and algebraic equations (DAE) rather than ordinary differential equations (ODE) (Nikravesh 1988, Blajer 1999, Flores and Seabra 2009).

Prior to integrate the system state variables, Eq. (141) is solved for $\ddot{\mathbf{q}}$ and λ by using any available numerical algorithm for linear equations, such those presented in section 4.

In the present work, the DAE are converted to ODE because the most frequently used numerical integration algorithms are useful in solving ODE (Shampine and Gordon 1975).

However, for a detailed discussion on DAE, the interested reader may consult the works by Petzold (1983) and Brenan et al. (1989).

The material presented below, relative to numerical integration of ODE, follows that of any undergraduate text on numerical analysis such as those by Conte and Boor (1981) and Atkinson (1989).

Generalities

The process of converting *n* second-order differential equations to 2*n* first-order equations was already presented in the previous section, which can be expressed by

$$\ddot{y}_1 = f(y_1, \dot{y}_1, t) \tag{145}$$

such that it can be written as the following system

$$\dot{y}_1 = y_2 \tag{146}$$

$$\dot{y}_2 = f(y_1, y_2, t) \tag{147}$$

The most popular and used numerical integration methods introduced in the vast thematic literature are Euler method, Runge-Kutta methods and Adams predictor-corrector methods. These methods have been known for many years, for instance, the Runge-Kutta methods have been known for more than an hundred years, but their potential was not fully realized until computers became available. These methods involve a step-by-step process in which a sequence of discrete points t^0, t^1, t^2, …, t^n is generated.

The discrete points may have either constant or variable spacing defined as $h^i=t^{i+1}-t^i$, where h^i is the step size for any discrete point t^i. At each point t^i, the solution $y(t^i)$ is approximated by a number y^i. Since no numerical method is capable of finding $y(t^i)$ exactly, the quantity

$$\varepsilon_g^i = \left| y(t^i) - y^i \right| \tag{148}$$

represents the global or total error at $t=t^i$. The total error consists of two components, the truncation error and the round-off error. The truncation error depends on the nature of the numerical algorithm used in computing y^i. The round-off error is due to the finite word length in a computer.

Generalities

The integration methods are called single step methods when they only require information on the current time step to advance to the next time step.

Euler and Runge-Kutta methods are single step methods. When information of the previous steps is used, the algorithm methods are called multistep methods, as it is the case of Adams predictor-corrector schemes.

The single step methods are self starting and they need a minimum amount of storage requirements. These methods require a larger number of function evaluations, for instance, four for the fourth-order Runge-Kutta method. Function evaluation is the name of the process by which, given t and y, the value of $\dot{y}$ is computed.

The multistep methods require a small amount of function evaluations, particularly if the time step is chosen so that the number of predictor-corrector iterations per step is kept below two or three. Moreover, error estimates are easily provided and step size adjustments can be performed with no difficulties. The multistep methods are not self starting and require the help of a single step scheme to start the integration process (Atkinson 1989).

Regardless of the numerical method used, the numerical task deals with the integration of an initial-value problem that can be written as

$$\dot{y}_1 = f(y,t) \tag{149}$$

with the initial condition $y(t^0)=y^0$ and where y is the variable to be integrated and function $f(t,y)$ is defined by the computational sequence of the algorithm selected.

Equation (149) has a solution $y(t)$. The initial value y^0 can be defined for any value of t^0, although it is often assumed that a transformation has been made so that $t^0=0$.

This does not affect the solution or method used to approximate the solution.

Euler's Method

It is known that the Euler integration method is one of the simplest integrators available. This approach may be sufficient in giving a very rough idea of the motion of multibody systems. This method solves differential equations in a single step as

$$y^{i+1} = y^i + hf(y^i, t) \tag{150}$$

where h is the integration step size $h=t^{i+1}-t^i$, for i a non-negative integer. This method implies that the next step of the state variable can be evaluated by using the current state variable.

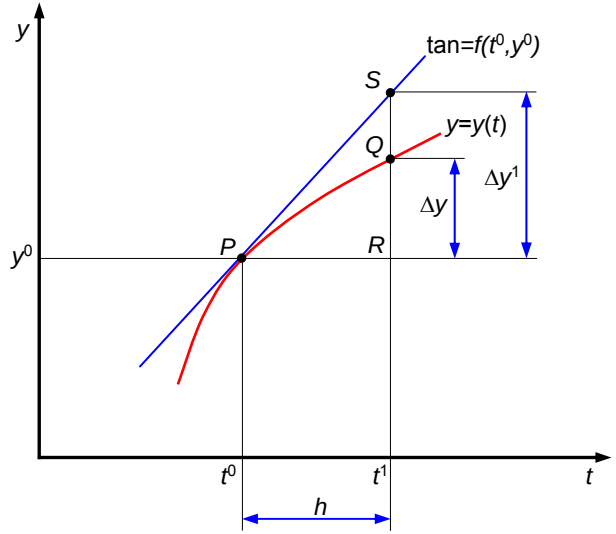


Fig. 52 Geometric interpretation of the Euler integration method

Euler's Method

The intuitive basis of the Euler method is illustrated in Fig. 52, in which the curve labeled $y=y(t)$ is the solution of the differential equation (149), which passes through point $P(t^0,y^0)$. It is desired to find the value of $y^1=y^0+\Delta y$ corresponding to $t=t^1$. In other words, the height RQ needs to be determined.

Although the position of the curve at every point is not known, its slope is equal to $f(t,y)$, which is simply the geometric interpretation of the differential equation. Thus, the slope of the tangent at point P is $\dot{y}^0 = f(t^0, y^0)$, which can be computed since y^0 and t^0 are both known.

If h is reasonable small, the tangent line PS should not deviate too much from the curve PQ, hence, the height RS (which by simple geometry is equal to $h\dot{y}^0$) should be an approximation to the required height RQ.

Thus, a first approximation to Δy is given by $\Delta y^1=RS=hf(t^0,y^0)$.

Assuming that the appropriate derivatives exist, then $y(t)$ can be expanded in a Taylor series about $t=t^i$ and the expression is evaluated at $t=t^{i+1}$, yielding (Atkinson 1989)

$$y(t^{i+1}) = y(t^i) + hf(t^i, y^i) + O(h^2) \tag{151}$$

From the analysis of Eq. (151), neglecting the higher-order terms, the discretization or local truncation error is given by

$$\varepsilon_l = O(h^2) \tag{152}$$

Euler's Method and Second Order Runge-Kutta Method

The order of a numerical integration method can be used to specify its accuracy and can be expressed using the local truncation error. Knowing that for a scalar equation of type

$$\varepsilon_l = O(h^{p+1}) \tag{153}$$

is said to be of p^{th} order, then it is clear that the Euler integration method is of first order. Thus, for highly oscillatory motion there are rapid changes in the derivatives of the function and if h is too large, then inaccuracies in the computation of the state variables are made (Nikravesh 1988).

In turn, the global truncation error at t^i can be evaluated as the difference between the actual and computed solution, in the absence of round-off error by the end of the simulation, that is

$$\varepsilon_g^i = \left| y(t^i) - y^i \right| \tag{154}$$

A more accurate integration method is the second-order Runge-Kutta algorithm, which can be expressed as

$$y^{i+1} = y^i + \frac{h}{2}(f_1 + f_2) \tag{155}$$

where

$$f_1 = f(t^i, y^i) \tag{156}$$

$$f_2 = f(t^i + h, y^i + hf_1) \tag{157}$$

Second Order Runge-Kutta Method

This approach is also known as the improved Euler method, modified trapezoidal method or the Heun method.

It should be noted that two function evaluations are required per time step, which in the case of multibody systems implies the solution of the equations of motion to obtain the accelerations twice at the given time step.

Figure 53 shows the geometric interpretation of the second-order Runge-Kutta method. This method is explicit in the measure that f^1 does not depend on f^2 and neither one depends on y^{i+1} (Jálon and Bayo 1994).

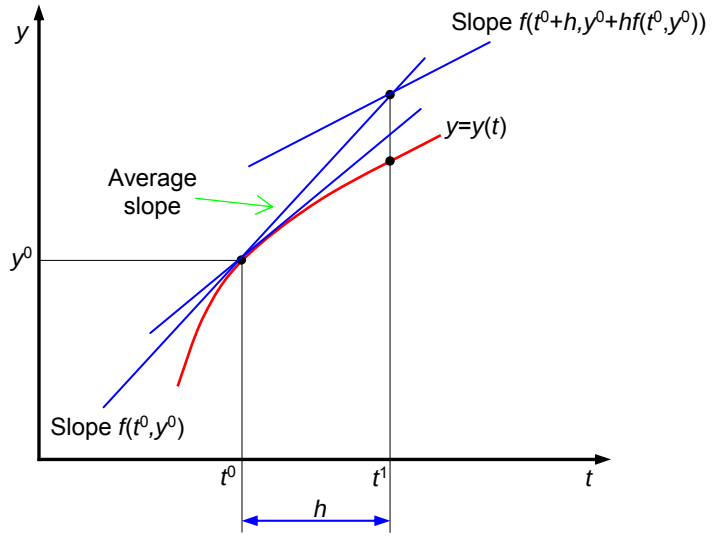


Fig. 53 Geometric interpretation of the second-order Runge-Kutta method

The local error of the second-order Runge-Kutta method is of order h^3, whereas that of Euler method is h^2. Thus, it is expected to be able to use a larger time step with the second-order Runge-Kutta method.

The price to pay for this is that it requires to evaluate the function $f(t,y)$ twice for each time step of the integration process.

Fourth Order Runge-Kutta Method

For larger time steps and for greater accuracy, the fourth-order Runge-Kutta integration method is most popular and widely used. This method is stable and, as a computer program, occupy relatively small amount of core storage. The fourth-order Runge-Kutta integration algorithm can be expressed by (Pina 1995)

$$y^{i+1} = y^i + hg \tag{158}$$

where

$$g = \frac{1}{6}(f_1 + 2f_2 + 2f_3 + f_4) \tag{159}$$

$$f_1 = f(t^i, y^i) \tag{160}$$

$$f_2 = f(t^i + \frac{h}{2}, y^i + \frac{h}{2} f_1) \tag{161}$$

$$f_3 = f(t^i + \frac{h}{2}, y^i + \frac{h}{2} f_2) \tag{162}$$

$$f_4 = f(t^i + h, y^i + hf_3) \tag{163}$$

This method is explicit because all f_i depend only on previous values already calculated. This algorithm is easy to implement in the measure that it only requires function evaluations, and it is self starting integrator scheme, which means that there is no need for any other algorithm or technique to start the integration process.

Fourth Order Runge-Kutta Method

Figure 54 illustrates the geometric interpretation of the fourth-order Runge-Kutta integration method. In this method four tangents are determined, being their average weighted according to Eqs. (159)-(163).

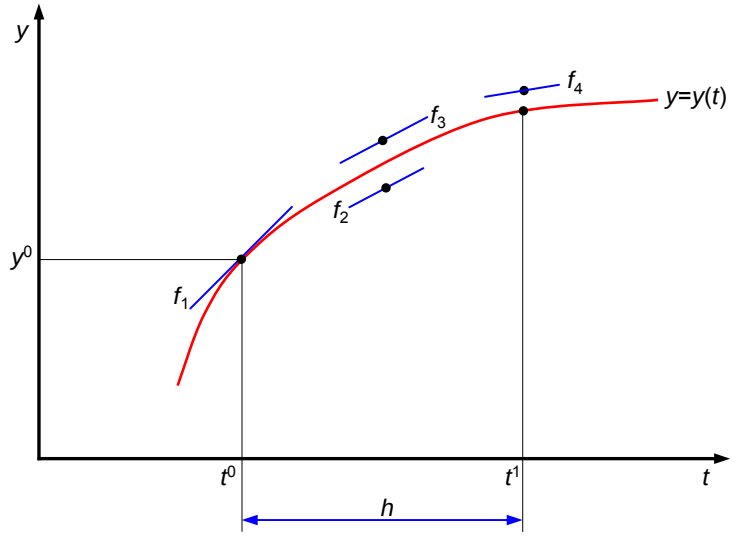


Fig. 54 Geometric interpretation of the fourth-order Runge-Kutta method

The standard fourth-order Runge-Kutta method does not provide an estimate of the local error, so that the user does not have way of knowing whether the time step being used is adequate. The local error of this method is of order h^5, which is relatively small even for larger time steps.

The major disadvantage of this method is that the function $f(t,y)$ needs to be evaluated four time at each time step. This method is less efficient than the multistep algorithms such as the Adams predictor-corrector schemes.

On some problems, Runge-Kutta method requires almost twice as much computing time (Conte and Boor 1981).

Fourth Order Runge-Kutta Method

Four the Euler and Runge-Kutta methods the next step value y^{i+1} is computed by using solely the current value y^i and time t^i, over a time range of $h=t^{i+1}-t^i$.

Multistep methods utilize information about the solution at more than one point.

The objective of the multistep methods is to automatically select the proper order and the proper time step size, which will minimize the amount of work required to achieve the specified accuracy for a given problem.

The multistep algorithms require only two function evaluation per step compared with four function evaluations per step with the fourth-order Runge-Kutta method, being, therefore, considerably faster and require less computation work.

Predictor-corrector methods provide an automatic error estimate at each time step, thus allowing the algorithm to select an optimum value of h for a required accuracy.

This type of approach is also better with respect to the propagation of error that it can use time steps more than twice as large.

Adams Predictor-Corrector Methods

In Adams predictor-corrector methods an explicit method is used to predict a value of y^{i+1}, while an implicit method corrects that value. The implicit corrects appear to be more stable and accurate than the explicit predictors and are both chosen to be of equal order.

The Adams-Bashforth predictor algorithm of fourth-order can be written as

$$y^{i+1} = y^i + \frac{h}{24}(55 f^i - 59 f^{i-1} + 37 f^{i-2} - 9 f^{i-3}) \tag{164}$$

where

$$f^i = f(t^i, y^i) \tag{165}$$

$$f^{i-j} = f(t^{i-j}, y^{i-j}) \quad (j=1, 2, 3) \tag{166}$$

The corresponding Adams-Moulton corrector algorithm can be expressed by

$$y^{i+1} = y^i + \frac{h}{24}(9 f^{i+1} + 19 f^i - 5 f^{i-1} + f^{i-2}) \tag{167}$$

where

$$f^i = f(t^i, y^i) \tag{168}$$

$$f^{i-j} = f(t^{i-j}, y^{i-j}) \quad (j=1, 2) \tag{169}$$

Adams Predictor-Corrector Methods

The major disadvantage of multistep methods is that they are not self starting.

Thus, in the fourth-order Adams predictor-corrector method four successive values of function evaluation at equally spaced points before instant of time t^i must be known.

These starting values must be obtained by some independent method, such as the Runge-Kutta method.

On the other hand, Adams predictor-corrector algorithms are more complicated to program in the measure that they require special techniques for starting and for doubling and halving the time step, and they be subject to numerical instability (Conte and Boor 1981, Pina 1995).

In short, the Adams methods, when being carefully use, are more efficient than any other method. To achieve this efficiency it is necessary to vary the time step and the order that are used.

Thus, it is necessary to estimate the errors that are incurred for various time steps and orders so as to make these decisions.

Advanced codes also attempt to detect abnormal situations such as discontinuities or certain types of instabilities and to deal with them in a reasonable way.

A detailed discussion on the Adams predictor-corrector implementation can be found in the textbooks by Shampine and Gordon (1975) and Jálon and Bayo (1994).

Gear Method

Gear (1971) developed a family of variable order stiffly-stable algorithms for the solution of stiff problems. A stiff system is referred to as any initial-value problem in which the complete solution consists of fast and slow components.

The stiffness can be produced by physical characteristics of the multibody systems, such as components with large differences in their masses, stiffness and damping.

However, in many other instances, stiffness is numerically induced due to either the discretization process, the large number of components and equations of motion, or sudden or accumulated violations in the constraint conditions.

The Gear algorithm of fourth-order can be expressed as

$$y^{i+1} = \frac{1}{25}(48y^{i} - 36y^{i-1} + 16y^{i-2} - 3y^{i-3} + 12hf^{i+1}) \tag{170}$$

where

$$f^{i+1} = f(t^{i+1}, y^{i+1}) \tag{171}$$

Since the Gear algorithm is an implicit multistep scheme, it is necessary to solve an implicit equation in each time step (Nikravesh 1988).

Finally, it should be highlighted that the choice of the integration time step is a very important issue, too small time steps lead to high computational time, and too large time steps induce instability of the calculus.

Therefore, it is extremely important to choose a reasonably small time step to obtain accurate results without unnecessary increasing the computation time (Nikravesh 2008).

Description of the Baumgarte's Method

In this section, the fundamentals of the Baumgarte stabilization method to control the violation of constraint are presented and discussed.

It is known that the initial conditions and the integration of the velocities and accelerations of a multibody system introduce numerical errors in the new positions and velocities computed.

These errors are due to the finite precision of the numerical methodologies and to the position and velocity constraint equations not appearing anywhere in the standard solution of the equations of motion.

Therefore, methods able to eliminate errors in the position or velocity equations or, at least, to keep such errors under control must be implemented. In order to keep the constraint violations under control, the Baumgarte stabilization method is considered here (Baumgarte 1972).

This method allows constraints to be slightly violated before corrective actions can take place, in order to force the violation to vanish. The objective of Baumgarte method is to replace the differential Eq. (140) by the following equation

$$\ddot{\mathbf{\Phi}} + 2\alpha\dot{\mathbf{\Phi}} + \beta^2\mathbf{\Phi} = \mathbf{0} \tag{172}$$

Equation (172) is a differential equation for a closed loop system in terms of kinematic constraint equations in which the terms $2\alpha\dot{\mathbf{\Phi}}$ and $\beta^2\mathbf{\Phi}$ play the role of control terms.

The principle of the method is based on the damping of acceleration of constraint violation by feeding back the position and velocity of constraint violations, as illustrated in Fig. 55, which shows open loop and closed loop control systems.

In the open loop systems $\mathbf{\Phi}$ and $\dot{\mathbf{\Phi}}$ do not converge to zero if any perturbation occurs and, therefore, the system is unstable.

Description of the Baumgarte's Method

Thus, using the Baumgarte approach, the equations of motion for a system subjected to constraints are stated in the following form

$$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_{\mathbf{q}}^{T} \\ \mathbf{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{g} \\ \boldsymbol{\gamma} - 2\alpha\dot{\mathbf{\Phi}} - \beta^{2}\mathbf{\Phi} \end{Bmatrix} \tag{173}$$

If α and β are chosen as positive constants, the stability of the general solution of Eq. (173) is guaranteed. Baumgarte (1972) highlighted that the suitable choice of the parameters α and β is performed by numerical experiments. Hence, the Baumgarte method has some ambiguity in determining optimal feedback gains.

Indeed, it seems that the value of the parameters is purely empiric, and there is no reliable method for selecting the coefficients α and β. The improper choice of these coefficients can lead to unacceptable results in the dynamics of the multibody systems (Nikravesh 1984, Flores and Seabra 2009).

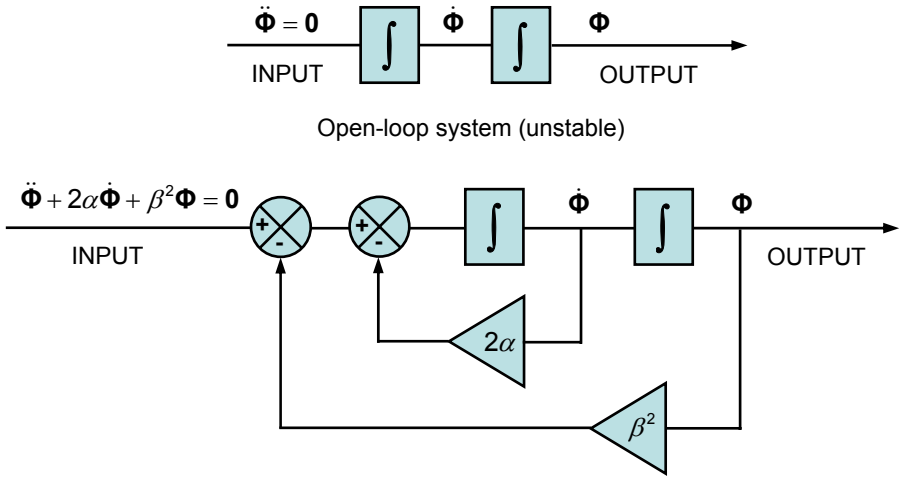


Fig. 55 Open loop and closed loop control systems

Description of the Baumgarte's Method

In what follows, some basic mathematical aspects dealing with the response of the Baumgarte method are analyzed. It is known that the characteristic or polynomial equation of the differential Eq. (172) can be written as (Zwillinger 1997, Polyanin and Zaitsev 2003)

$$s^2 + 2\alpha s + \beta^2 = 0 \tag{174}$$

whose roots are given by

$$s_{1,2} = -\alpha \pm \sqrt{\alpha^2 - \beta^2} \tag{175}$$

Based on mathematical knowledge, in the case of equations such as Eq. (175), 3 different situations can occur

$$\alpha^2 - \beta^2 > 0 \tag{176}$$

$$\alpha^2 - \beta^2 = 0 \tag{177}$$

$$\alpha^2 - \beta^2 < 0 \tag{178}$$

In order to better understand what are the consequences in terms of the systems stability these three cases are studied in detail. In the first case, when $\alpha^2 - \beta^2 > 0$, the characteristic equation has two real roots

$$s_1 = -\alpha + \sqrt{\alpha^2 - \beta^2} \tag{179}$$

$$s_2 = -\alpha - \sqrt{\alpha^2 - \beta^2} \tag{180}$$

Description of the Baumgarte's Method

Thus, the solution of Eq. (172) can be written as (Zwillinger 1997)

$$\mathbf{\Phi}(t) = c_1 e^{\left(-\alpha+\sqrt{\alpha^2-\beta^2}\right)t} + c_2 e^{\left(-\alpha-\sqrt{\alpha^2-\beta^2}\right)t} \tag{181}$$

Equation (181) incorporates the physical system properties and it is usually called system weight function. This system weight function should satisfy the following conditions (Polyanin and Zaitsev 2003)

$$\mathbf{\Phi}(0) = 0 \tag{182}$$

$$\dot{\mathbf{\Phi}}(0) = 0 \tag{183}$$

Based on these two conditions it is possible to study the response of Eq. (181). Thus, considering the first condition given by Eq. (182) yields

$$c_1 = -c_2 \tag{184}$$

Now, taking Eqs. (183) and (184) results in

$$c_1 = \frac{1}{2\sqrt{\alpha^2-\beta^2}} \tag{185}$$

$$c_2 = -\frac{1}{2\sqrt{\alpha^2-\beta^2}} \tag{186}$$

Description of the Baumgarte's Method

Finally, Eq. (181) can be re-written as

$$\Phi(t) = \frac{1}{2\sqrt{\alpha^2 - \beta^2}} e^{\left(-\alpha+\sqrt{\alpha^2-\beta^2}\right)t} - \frac{1}{2\sqrt{\alpha^2 - \beta^2}} e^{\left(-\alpha-\sqrt{\alpha^2-\beta^2}\right)t} \tag{187}$$

From Eq. (187), it can be observed that it is stable when the 2 following equations are simultaneously verified

$$-\alpha + \sqrt{\alpha^2 - \beta^2} < 0 \tag{188}$$

$$-\alpha - \sqrt{\alpha^2 - \beta^2} < 0 \tag{189}$$

The behavior of Eq. (187) for different values of α and β parameters is shown in Fig. 56, where after the initial overshoot the system response tends to zero more or less quickly, depending on the values of α and β.

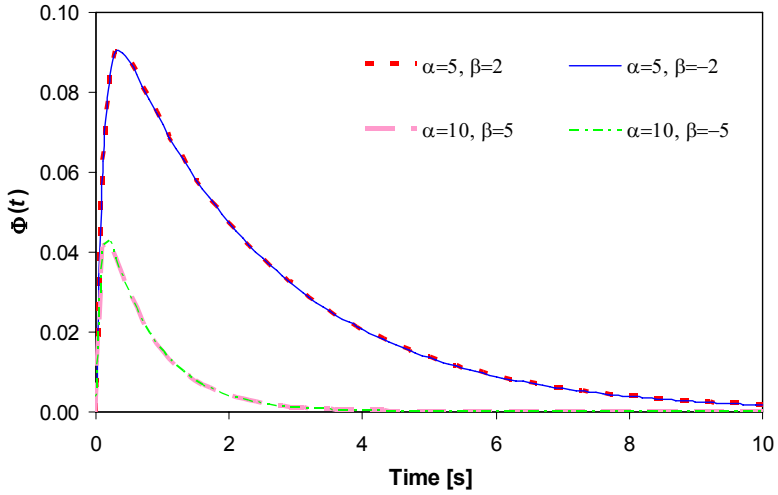


Fig. 56 Behavior of Eq. (187) for different values of α and β

Description of the Baumgarte's Method

Considering now the second case expressed by Eq. (177), the characteristic equation has a double root expressed as

$$s = -\alpha \tag{190}$$

In this situation, the solution of Eq. (172) is given by

$$\mathbf{\Phi}(t) = (c_1 t + c_2)e^{-\alpha t} \tag{191}$$

Taking again the two system weight functions given by Eqs. (182) and (183) yields

$$c_1 = 1 \tag{192}$$

$$c_2 = 0 \tag{193}$$

Finally, Eq. (191) is re-written as

$$\mathbf{\Phi}(t) = te^{-\alpha t} \tag{194}$$

From Eq. (194) it can be concluded that it is stable if α>0.

Figure 57 illustrates the response of Eq. (194) for different values of α, being clear that when the parameter α increases the overshoot decreases.

Description of the Baumgarte's Method

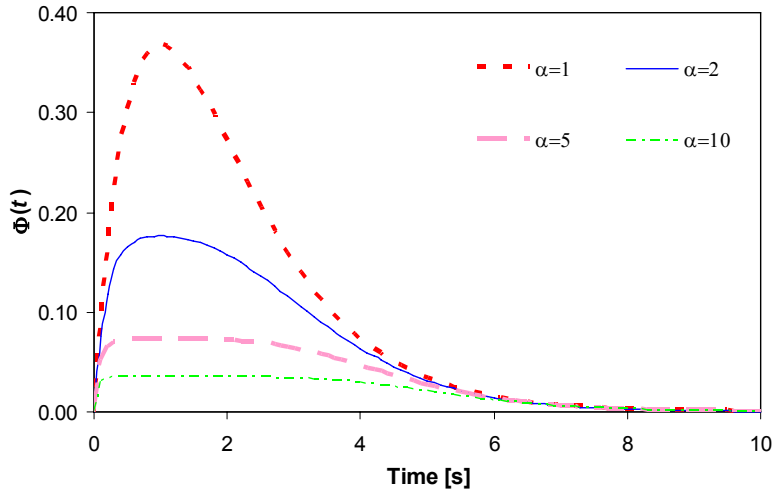


Fig. 57 Behavior of Eq. (194) for different values of α

Finally, in the third case, when $\alpha^2 - \beta^2 < 0$, the characteristic equation has two complex and conjugate roots

$$s_1 = -\alpha + i\sqrt{\alpha^2 - \beta^2} \tag{195}$$

$$s_2 = -\alpha - i\sqrt{\alpha^2 - \beta^2} \tag{196}$$

In this case, the solution of Eq. (172) is expressed by

$$\mathbf{\Phi}(t) = e^{-\alpha t}\left(c_1 \cos\sqrt{\beta^2 - \alpha^2}\,t + c_2 \sin\sqrt{\beta^2 - \alpha^2}\,t\right) \tag{197}$$

Description of the Baumgarte's Method

Considering the system weight functions expressed by Eqs. (182) and (183) yields

$$c_1 = 0 \tag{198}$$

$$c_2 = \frac{1}{\sqrt{\beta^2 - \alpha^2}} \tag{199}$$

Finally, Eq. (197) can be re-written as

$$\Phi(t) = \frac{1}{\sqrt{\beta^2 - \alpha^2}} e^{-\alpha t} \sin \sqrt{\beta^2 - \alpha^2}\, t \tag{200}$$

Figure 58 illustrates the response of Eq. (200) for different values of α and β parameters. The system's frequency decreases when the values of α and β increase and the system stabilizes more quickly.

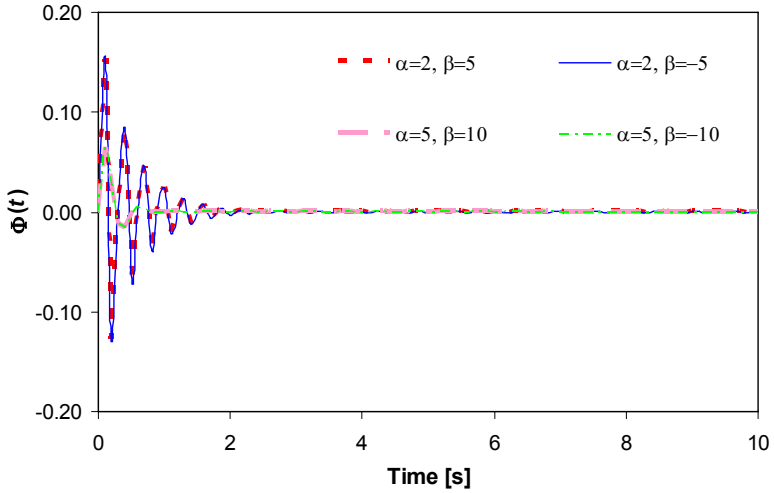


Fig. 58 Behavior of Eq. (200) for different values of α and β

Techniques to Select the Baumgarte Parameters

In this section two different approaches to help in the selection of the Baumgarte parameters are presented. The first and simplest way to evaluate the Baumgarte parameters consists of expanding in Taylor's series the constraint equation and neglecting the terms of order higher than two. Thus, it is possible to write

$$\mathbf{\Phi}(t+h) = \mathbf{\Phi}(t) + \dot{\mathbf{\Phi}}(t)h + \ddot{\mathbf{\Phi}}(t)\frac{h^2}{2} \tag{201}$$

where *h* represents the time step.

Considering that function $\mathbf{\Phi}$ is null at instant *t*+*h*, then Eq. (201) can be re-written as

$$\ddot{\mathbf{\Phi}}(t) + \frac{2}{h}\dot{\mathbf{\Phi}}(t) + \frac{2}{h^2}\mathbf{\Phi}(t) = 0 \tag{202}$$

By comparing and analyzing Eqs. (172) and (202) the mathematical relation for Baumgarte parameters and time step can be expressed by

$$\alpha = \frac{1}{h} \tag{203}$$

$$\beta = \frac{\sqrt{2}}{h} \tag{204}$$

From Eqs. (203) and (204) it can be observed that with this technique the Baumgarte parameters are inversely proportional to the time step. This approach is quite simple, very easy to implement in any general code and works reasonably well from the computational view point.

However, this procedure can lead to some numerical instability which ultimately produces incorrect results when the time step is too small, because the damping terms dominate the numerical value of Eq. (172) and make the system to become stiff.

Techniques to Select the Baumgarte Parameters

When a fixed integration step size algorithm is used and the time step is not very large, this approach indeed damps out the constraint violations faster than an assigned constant value of parameters α and β.

Even in most problems of midlevel of complexity, in particular when including contact-impact analysis, the constant integration time step may fail to converge because of the integration error accumulation.

Thus, a better error-control algorithm must be employed, that is, algorithm in which the time step and order vary with the purpose to effectively control the integration error.

Therefore, a more sophisticated methodology should be considered to select an appropriate set of α and β parameters.

The presented methodology is based on the stability analysis procedure in digital control theory.

When a numerical algorithm is employed to solve the equations of motion, in fact the differential equations are transformed into difference equations, which are then solved.

The system of difference equations is characterized by fixed and discrete time intervals of the system variables.

Several integration methods lead to different solving procedures and the stability analysis have to be applied to the resulting difference equations rather than the original differential equations.

For this, it can be observed that the choice of the α and β parameters depends on the integration method used.

In order to better understand the influence of the integration method on the selection of the Baumgarte parameters, the Euler method is chosen for this purpose due to its simplicity.

It should be highlighted that this explicated method is one of the first and simplest integration methods available in the vast thematic literature (Hildebrand 1974, Leader 2004).

Techniques to Select the Baumgarte Parameters

Applying the Laplace transform technique to a first-order differential equation yields (Cochin and Cadwallender 1997)

$$sY(s) = F(s) \tag{205}$$

where *s* is the operator of Laplace domain.

On the other hand, when Euler integration method is used, the numerical solution of a first-order differential equation can be written as

$$y^{i+1} = y^i + hf^i \tag{206}$$

in which the superscript represents the numerical solution at the corresponding time step and *h* is the integration time step.

Since Eq. (206) is a difference equation, that is, a discrete data function, the *Z* transform technique must be used to study it. In fact, the *Z* transform theory extends the power and convenience of the calculus to the realm of discrete data systems, rather than Laplace method, which deals with continuous systems (Cochin and Cadwallender 1997).

Thereby, the Z transform of Eq. (206) results in

$$zY(z) = Y(z) + hF(z) \tag{207}$$

where *z* is the *Z* transform variable.

Techniques to Select the Baumgarte Parameters

Re-arranging Eq. (207) yields

$$\frac{F(z)}{Y(z)} = \frac{z-1}{h} \tag{208}$$

Analyzing Eqs. (205) and (208), striking resemblances between Laplace and Z transform techniques results in

$$s = \frac{z-1}{h} \tag{209}$$

This means that the substitution of Eq. (209) in any $F(s)/Y(s)$ yields a $F(z)/Y(z)$ based on the Euler integration method. Considering now Eq. (172), the corresponding characteristic equation is

$$s^2 + 2\alpha s + \beta^2 = 0 \tag{210}$$

Equation (210) suggests that if α and β are greater than zero, the system is be stable. However, Eq. (210) is not adequate to select α and β parameters as it was shown in the previously. In order to select the appropriate values of the parameters α and β, the response of the second order characteristic equation (210) for different locations of its roots in the z-plane must be determined first.

On the other hand, it is known that on the s-plane, the region of stability is the left half-plane, being the region of stability on the z-plane evaluated from the definition, $z=e^{ts}$. Letting $s = \sigma + j\omega$, it is possible to write (Zwillinger 1997)

$$z = e^{t(\sigma + j\omega)} = e^{\sigma t} \angle \omega t \tag{211}$$

since $1\omega t = \cos \omega t + j \sin \omega t.$

Techniques to Select the Baumgarte Parameters

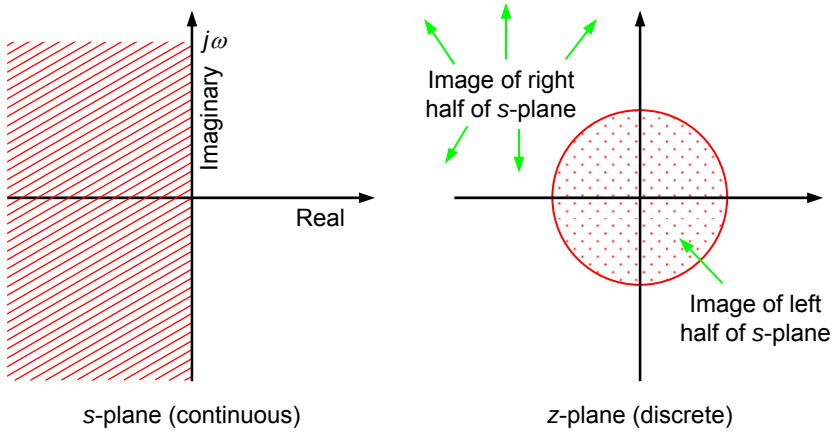


Fig. 59 Mapping regions of the *s*-plane into the *z*-plane

As a consequence, each region of the *s*-plane can be mapped into a corresponding region on the *z*-plane, as it is shown in Fig. 59.

Points that have positive values of σ are located in the right half *s*-plane. From Eq. (211) it can be considered that the magnitudes of that mapped points are $e^{ts}>1$. Consequently, points in the right half of the *s*-plane map into outside the unit circle on the *z*-plane.

Points on the $j\omega$-axis have zero values of σ and yield points on the *z*-plane with magnitude equal 1, that is, the unit circle.

Finally, points on the *s*-plane that yield negative value of *s* (left half plane roots) map into the inside of the unit circle on the *z*-plane.

A system is stable if all roots of the characteristic equation are inside the unit circle on the *z*-plane, that is, $|z|<1$. Conversely, a system is said to be unstable when the roots are outside the unit circle, that is, $|z|>1$. When a system has its roots on the unit circle, $|z|=1$, is called as marginally stable.

Techniques to Select the Baumgarte Parameters

In order to study the stability Eq. (210), let consider Eq. (209) yielding the characteristic equation in terms of the *z*-plane as

$$z^2 + (2\alpha h - 2)z + (\beta^2 h^2 - 2\alpha h + 1) = 0 \tag{212}$$

Table 1 shows the stability analysis of Eq. (212) for a set of different values of parameters α and β, when the time step is considered to be fixed and equal to 0.001 s.

Tab.1 Stability analysis of the Euler integration method

α	β	$\lvert z \rvert$	$\angle z$	Stability
-5	-5	1.005	0.0º	Diverge
5	5	0.995	0.0º	Converge without oscillation
20	20	0.980	0.0º	Converge fast without oscillation
10	14	0.990	1.3º	Converge with oscillation
1000	1414	0.990	90.0º	Converge with oscillation
0.8	2100	2.326	64.6º	Diverge

Equation (212) shows that α, β and h can influence the location of the roots and, hence, the dynamic system response.

In order to have a criterion to help in the selection of the α and β parameters independently of the time step, let consider two additional coefficients $\underline{\alpha}$ and $\underline{\beta}$ defined by, $\underline{\alpha} = \alpha h$ and $\underline{\beta} = \beta^2 h^2$.

The relation between the $\underline{\alpha}$ and $\underline{\beta}$ coefficients for the Euler integration method is shown in the plot of Fig. 60, being easy to identify the stability region as function of the Baumgarte parameters.

Techniques to Select the Baumgarte Parameters

The relation between the α and β coefficients for the Euler integration method is shown in the plot of Fig. 60, being easy to identify the stability region as function of the Baumgarte parameters.

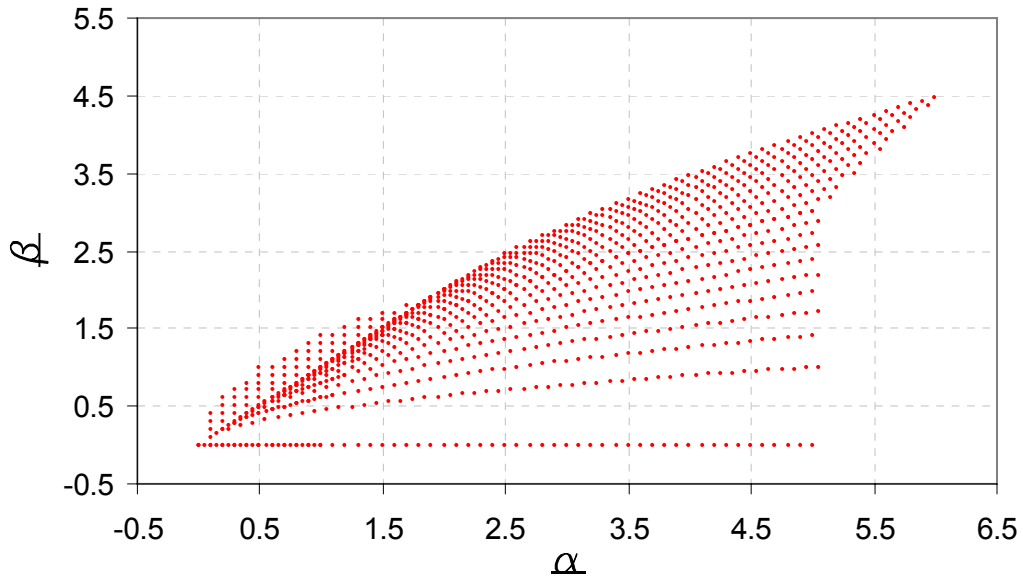


Fig. 60 Stability region in the α-β plane for the Euler method

Example of Application

A human body model is used as a numerical example to study the influence of the Baumgarte stabilization method on the control of the constraint violations. Figure 61 shows a schematic representation of this biomechanical model developed under the framework of multibody methodologies. The model is composed by seven rigid bodies, being six of them relative to the locomotor system, i.e., the two lower-limbs, and one that represents the main upper body segments here denominated by HAT, acronym for head, arms and trunk.

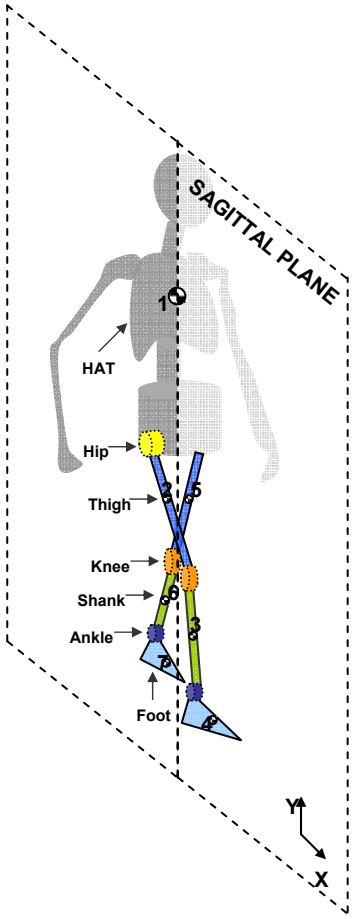


Fig. 61 Biomechanical multibody model of a human body on the sagittal plane

Example of Application

The anthropometric description of the seven anatomical segments considered and their corresponding body numbers are listed in Table 2 (Meireles 2007).

In this biomechanical model, the anatomical segments are connected by six revolute joints, which results in twelve kinematic constraints in the system.

These six revolute joints corresponds to the principal articular human joints, namely right hip, right knee, right ankle, left hip, left knee and left ankle.

These six revolute joints reduce the *DOF* of the system to nine, which correspond to six rotations about revolute joints axis, plus two translations and one rotation of the main body (HAT).

Therefore, nine guiding constraints are included to drive the system according to the *DOF* mention previously.

In other words, three guiding constraints are associated with the HAT body to guide the x, y and ϕ coordinates of its center of mass.

Additionally, a guiding constraint is allocated to each relative *DOF* of the locomotor system in order to restrain the relative rotation of the adjacent bodies.

The kinematic trajectories associated with the guiding constraints are obtained using cubic spline interpolation techniques.

The motion data utilized to derive the mathematical expressions by interpolating the coordinates along time is based on the Winter's work (2009).

These data represent a time period of 0.957 seconds related to a human gait cycle of normal cadence.

The first 0.4 seconds of simulation corresponds to the swing phase of gait cycle, and the remaining period of simulation corresponds to stance phase.

Example of Application

Tab. 2 Anthropometric data for each anatomical segment of the biomechanical human model

Segment	Description	Length [m]	CM Proximal Loc. [m]	Mass [Kg]	Moment of Inertia [Kg.m^2]
1	½ HAT	0.2575	-0.0355	19.2213	1.03923
2	Right Thigh	0.3141	0.1360	5.67000	0.05836
3	Right Shank	0.4081	0.1840	2.63655	0.04005
4	Right Foot	0.1221	0.0610	0.82215	0.00273
5	Left Thigh	0.3141	0.1360	5.67000	0.05836
6	Left Shank	0.4081	0.1840	2.63655	0.04005
7	Left Foot	0.1221	0.0610	0.82215	0.00273

Besides the kinematic constraints, external applied forces that act at the feet are also considered in this study as input data.

These forces are included to simulate the reaction forces between the feet and the ground.

The kinetic data presented in Winter (2009), namely the force magnitude and the coordinates of the point of application, is utilized in this study.

In each instant of simulation, the components of the external applied forces, designated by F_x and F_y, are transferred to the center of mass of each foot.

These transferred forces and moments are then added to the generalized force vector.

Example of Application

The rheonomic constraints, such as the guiding constraints, are not susceptible to be violated during a dynamic simulation, since they define explicitly the position of the bodies at each time step.

In turn, the scleronomic constraints, such as the revolute joints, are usually subjected to constraint violations when moderate or long time simulations are performed.

The first cause of constraints violations in multibody dynamics relies on the set of the initial conditions. In fact, the initial configuration given to the system can be inaccurate or not well-defined.

These inconsistencies lead to constraint violations even during the first instant of the simulation, and tends to increase with the time.

Hence, prior to run dynamic simulations is recommended to perform a previous kinematic analysis in order to correct the initial conditions of the system, namely the positions and velocities of the bodies (Silva and Ambrósio, 2002).

In this study, a pre-kinematic analysis was performed with this purpose. Thus, the risk of having constraint violations due to the inaccuracy of the initial conditions is avoid or at least minimized.

The effect of the application of the Baumgarte stabilization method is demonstrated throughout this work using the developed biomechanical model. For this purpose, two forward dynamic simulations are performed using the fourth-order Runge-Kutta method as numerical integrator, being the time step equal to 0.00145 s.

The first simulation corresponds to the direct integration method, which means that the stabilization method is not considered. In the second simulation, the Baumgarte stabilization method is used, being the Baumgarte parameters, α and β, both equal to 5. The obtained results for these two computational simulations are also compared with the Winter's data (2009).

These results are plotted in Figure 62, where it is visible the effect of the Baumgarte stabilization method on the control of the violations of the position constraints of the four bodies, namely HAT, thigh, shank and foot.

Example of Application

From the analysis of Figure 62, it can be observed that when the direct integration method is used the violation of constraints significantly grow with time, which produces unacceptable results, even for a relatively short simulation. In turn, when the Baumgarte method is utilized, the system response is clearly different, as it is illustrated in the plots of Figure 62. In last case, the outcomes are close to the Winter's data (2009), which represents a good correlation between the predicted results and those published by Winter (2009).

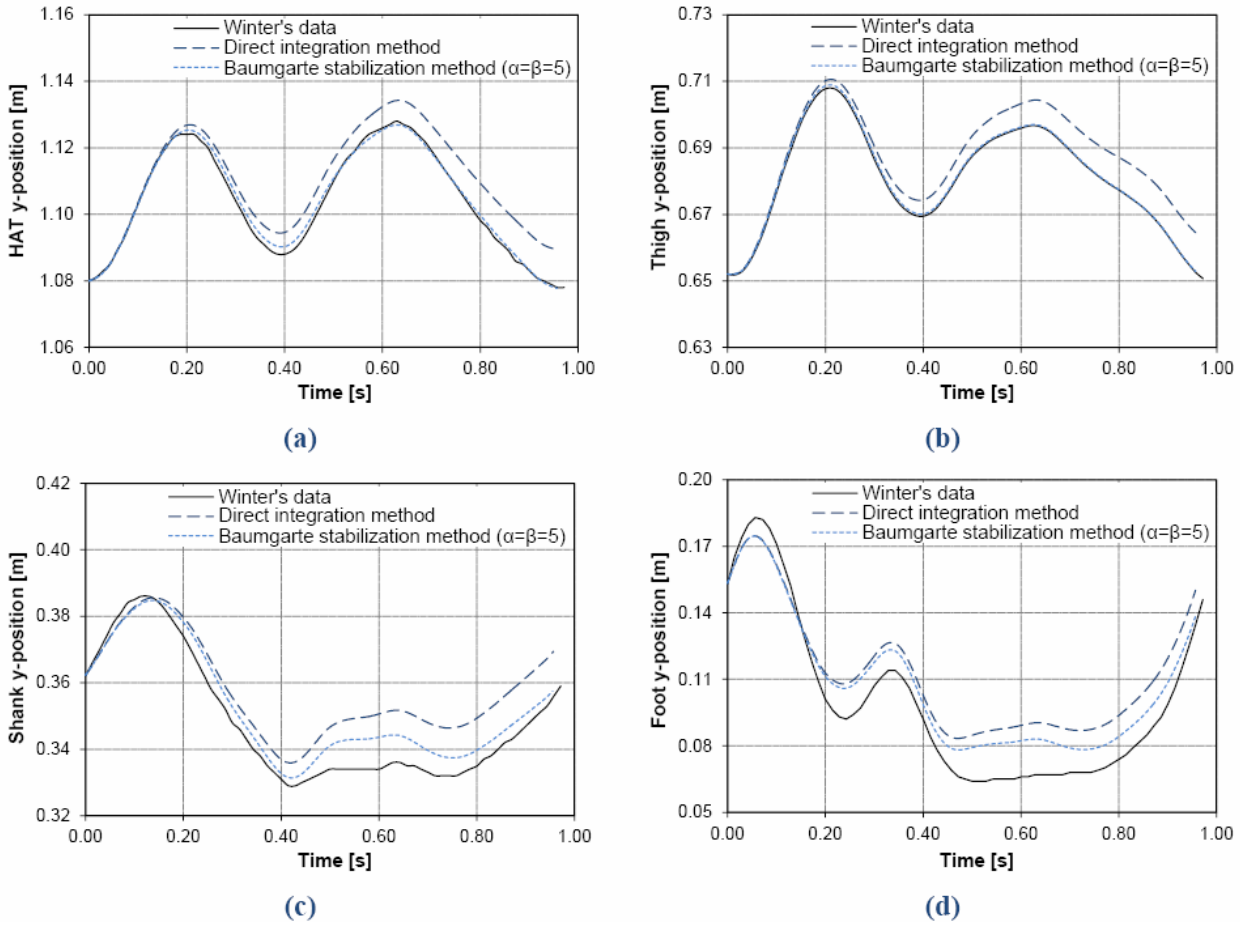


Fig. 62 Influence of the Baumgarte parameters, on the *y*-position of the biomechanical model bodies: (a) HAT; (b) Thigh; (c) Shank; (d) Foot.

List of References

Ambrósio J, Veríssimo P (2009) Sensitivity of a vehicle ride to the suspension bushing characteristics. Journal of Mechanical Science and Technology, 23:1075-1082.

Ambrósio JAC, Neto MA, Leal RP (2007) Optimization of a complex flexible multibody systems with composite materials. Multibody System Dynamics, 18:117-144.

Amirouche FML (1992), Computational methods for multibody dynamics. Prentice Hall, Englewood Cliffs, New Jersey

Arabyan A, Wu F (1998) An improved formulation for constrained mechanical systems. Multibody Systems Dynamics 2(1):49-69.

Atkinson KA (1989) An introduction to numerical analysis, Second edition, John Wiley and Sons. New York.

Baumgarte J (1972) Stabilization of constraints and integrals of motion in dynamical systems. Computer Methods in Applied Mechanics and Engineering 1:1-16.

Bayo E, Jálon JG, Serna AA (1988) Modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems. Computer Methods in Applied Mechanics and Engineering 71:183-195.

Blajer W (1995) An orthonormal tangent space method for constrained multibody systems. Computer Methods in Applied Mechanics and Engineering 121:45-57.

Blajer W (1999) Elimination of constraint violation and accuracy improvement in numerical simulation of multibody systems. Proceedings of EUROMECH Colloquium 404, Advances in Computational Multibody Dynamics, IDMEC/IST, edited by J. Ambrósio and W. Schiehlen, Lisbon, Portugal, September 20-23, 769-787.

Brenan KE, Campbell SL, Petzold LR (1989) Numerical solution of initial-value problems in differential-algebraic equations. Elsevier Science Pub. Co. New York.

Chace MA (1967) Analysis of the time-dependence of multi-freedom mechanical systems in relative coordinates. Journal of Engineering for Industry, 89:119-125.

List of References

Chapra SC, Canale RP (1989) Numerical methods for engineers. 2nd Ed., McGraw-Hill.

Cochin I, Cadwallender W (1997) Analysis and design of dynamic systems. 3rd edition, Addison Wesley, New Jersey.

Conte SD, Boor C (1981) Elementary numerical analysis: An algorithmic approach. Third edition, McGraw-Hill, Singapore.

Dahlquist G, Björck A (1974) Numerical methods. Prentice-Hall, New Jersey.

Eich-Soellner E, Führer C (1998) Numerical methods in multibody dynamics. Teubner-Verlag Stuttgart, Germany.

Fisette P, Vaneghem B (1996) Numerical integration of multibody system dynamic equations using the coordinate method in an implicit Newmark scheme. Computer Methods in Applied Mechanics and Engineering 135:85-105.

Flores P (2006a) Modelação de restrições cinemáticas em sistemas de corpos múltiplos. School of Engineering, University of Minho. Guimarães, Portugal.

Flores P (2006b) Análise cinemática de sistemas de corpos múltiplos. School of Engineering, University of Minho. Guimarães, Portugal.

Flores P (2008) Dinâmica de mecanismos: forças existentes nos mecanismos. School of Engineering, University of Minho. Guimarães, Portugal.

Flores P, Ambrósio J, Claro JCP, Lankarani HM (2006) Influence of the contact-impact force model on the dynamic response of multibody systems. Proceedings of the Institution of Mechanical Engineers, Part-K Journal of Multi-body Dynamics, 220(1):21-34.

Flores P, Ambrósio, J, Claro, JCP, Lankarani, HM (2008) Kinematics and dynamics of multibody systems with imperfect joints: models and case studies. Vol. 34, Berlin Heidelberg New-York, Springer-Verlag.

List of References

Flores P, Claro JCP (2007a) A systematic and general approach to kinematic position errors due to manufacturing and assemble tolerances. Proceedings of ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Las Vegas, Nevada, USA, September 4-7, 7p.

Flores P, Claro JCP (2007b) Cinemática de mecanismos. Almedina, Coimbra, Por-tugal.

Flores P, Seabra E (2009) Influence of the Baumgarte parameters on the dynamics response of multibody mechanical systems. Dynamics of Continuous, Discrete and Impulsive Systems, Series B: Applications and Algorithms, Vol. 16(3): 415-432.

Gear CW (1971) Numerical initial value problems in ordinary differential equations. Prentice-Hall Englewood Cliffs, New Jersey.

Gear CW (1981) Numerical solution of differential-algebraic equations. IEEE Transactions on Circuit Theory CT-18:89-95.

Haug EJ (1989) Computer-aided kinematics and dynamics of mechanical systems - Volume I: basic methods. Allyn and Bacon, Boston, Massachusetts.

Hildebrand FB (1974) Introduction to numerical analysis. 2nd edition, McGraw-Hill.

Huston RL (1990) Multibody dynamics. Butterworth-Heinemann, Boston, Massachusetts.

Jálon JG (2007) Twenty-five years of natural coordinates. Multibody System Dynamics, 18:15-33.

Jálon JG, Bayo E (1994) Kinematic and dynamic simulations of multibody systems: the real-time challenge. Springer Verlag, New York.

Leader JJ (2004) Numerical analysis and scientific computation Addison Wesley.

Meireles F (2007) Kinematics and dynamics of biomechanical models using multibody systems methodologies: A computational and experimental study of human gait. MSc Dissertation, University of Minho, Guimarães, PT.

List of References

Müller A (2009) Generic mobility of rigid body mechanisms. Mechanism and Machine Theory, 44(6):1240-1255.

Neto MA, Ambrósio J (2003) Stabilization methods for the integration of DAE in the presence of redundant constraints. Multibody System Dynamics 10:81-105.

Nikravesh PE (1984) Some methods for dynamic analysis of constrained me-chanical systems: A survey. In Computer-Aided analysis and Optimization of Mechanical System Dynamics (edited by E.J. Haug), Springer Verlag, Berlin, Germany, 351-368.

Nikravesh PE (1988) Computer-aided analysis of mechanical systems. Prentice Hall, Englewood Cliffs, New Jersey.

Nikravesh PE (2007) Initial condition correction in multibody dynamics. Multi-body System Dynamics 18: 107-115.

Nikravesh PE (2008) Planar multibody dynamics: formulation, programming, and applications. CRC Press, London.

Orlandea N, Chace MA, Calahan DA (1977) A sparsity oriented approach to the dynamic analysis and design of mechanical systems - Part 1 and 2. Journal of Engineering for Industry, 99:773-784.

Paul B, Krajcinovic D (1970) Computer analysis of machines with planar motion, Part 1 - Kinematics, Part 2 - Dynamics. Journal of Applied Mechanics, 37:697-712.

Petzold LR (1983) A description of DASSL: A differential/algebraic system solver. Scientific Computing edited by R. Stepleman et al., North-Holland Pub. Co. 65-68.

Pina H (1995) Métodos numéricos. McGraw-Hill, Lisboa, Portugal.

Polyanin AD, Zaitsev VF (2003) Handbook of exact solutions for ordinary differential equations. 2nd edition, Chapman & Hall/CRC Press, Boca Raton.

List of References

Pombo J, Ambrósio J (2008) Application of a wheel-rail contact model to railway dynamics in small radius curved tracks. Multibody System Dynamics, 19:91-114.

Reuleaux F (1963) The kinematics of machinery. Dover, New York.

Schiehlen W (1990) Multibody systems handbook. Springer-Verlag, Berlin, Ger-many.

Seabra E, Flores P, Silva JF (2007) Theoretical and experimental analysis of an industrial cutting file machine using multibody systems methodology. Proceedings of ECCOMAS Thematic Conference Multibody Dynamics 2007, edited by C.L. Bottasso, P. Masarati, L. Trainelli, Milan, Italy, 25-28 June, 12p.

Shabana AA (1989) Dynamics of multibody systems. John Wiley and Sons, New York.

Shampine L, Gordon M (1975) Computer solution of ordinary differential equations: The initial value problem. Freeman, San Francisco, California.

Sheth PN, Uicker JJ (1971) IMP (Integrated Mechanism Program): a computer-aided design analysis system for mechanisms and linkages. Journal of Engineering for Industry, 94(2):454-464.

Shigley JE, Uicker JJ (1995) Theory of machines and mechanisms. McGraw Hill, New York.

Silva M, (2003) Human motion analysis using multibody dynamics and optimization tools. PhD Dissertation, Technical University of Lisbon, Lisbon, Portugal.

Silva MPT, Ambrósio JAC (2002) Kinematic data consistency in the inverse dynamic analysis of biomechanical systems. Multibody System Dynamics, 8:219-239.

Späth H (1995) One dimensional spline interpolation algorithms. AK Peters, Wellesley, Massachusetts.

Tseng F-C, Ma Z-D, Hulbert GM (2003) Efficient numerical solution of con-strained multibody dynamics systems. Computer Methods in Applied Mechanics and Engineering 192:439-472.

List of References

Wehage RA, Haug EJ (1982) Generalized coordinate partitioning for dimension reduction in analysis of constrained systems. Journal of Mechanical Design, 104:247-255.

Weijia Z, Zhenkuan P, Yibing W (2000) An automatic constraint violation stabilization method for differential/algebraic equations of motion in multibody sys-tem dynamics. Applied Mathematics and Mechanics 21(1):103-108.

Yoon S, Howe RM, Greenwood DT (1994) Geometric Elimination of Constraint Violations in Numerical Simulation of Lagrangian Equations. Journal of Mechanical Design 116:1058-1064.

Zhu W-H, Piedboeuf J-C, Gonthier Y (2006) A dynamics formulation of general constrained robots. Multibody System Dynamics, 16:37-54.

Zwillinger D (1997) Handbook of differential equations. 3rd edition, Academic Press, Boston.